Multi-viewpoint systems for 3-D visual communication

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.F. Wakker,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 9 oktober 2000 om 16:00 uur

door

Peter-André REDERT

elektrotechnisch ingenieur,
geboren te Vlaardingen.

Dit proefschrift is goedgekeurd door de promotor:

Prof.dr.ir. J. Biemond

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof.dr.ir. J. Biemond	Technische Universiteit Delft, promotor
Dr. E.A. Hendriks	Technische Universiteit Delft, toegevoegd promotor
Prof.dr.ir. A.W. Heemink	Technische Universiteit Delft
Prof.dr.ir. L.J. van Vliet	Technische Universiteit Delft
Prof.dr.ir. P.P.L. Regtien	Universiteit Twente
Prof.dr. A. Sarti	Politecno di Milano, Italië
Prof.dr. A.K. Katsaggelos	Northwestern University, USA

Multi-viewpoint systems for 3-D visual communication
Redert, Peter-André
Ph.D. Thesis Delft University of Technology – with ref. – with Summary in Dutch

Cover design ‘Businessman captured in 3-D’ by Jan-Willem Jekel
Printed by Universal Press, Veenendaal

ISBN 90-901-3985-0

Copyright © 2000 by P.A. Redert

All rights reserved. No part of this thesis may be reproduced, in any form or by any means, without permission in writing from the author.

Contents

1 INTRODUCTION......1

1.1 HUMAN VISION AND TECHNOLOGY......1
1.2 SYSTEMS FOR 3-D VISUAL COMMUNICATION......2
1.3 SCOPE OF THE THESIS......10
1.4 THESIS OUTLINE......14

2 STEREO CAMERA MODELS......17

2.1 INTRODUCTION......17
2.2 GEOMETRIC NOTATION......19
2.3 A GENERAL STEREO CAMERA MODEL......19
2.4 LESS COMPLEX MODELS......34
2.5 TRIANGULATION......41
2.6 PROPERTIES OF THE ACQUIRED SCENE......44
2.7 CONCLUSIONS......53

3 STEREO CAMERA CALIBRATION......55

3.1 INTRODUCTION......55
3.2 CALIBRATION OBJECT......58
3.3 MARKER DETECTION......59
3.4 MARKER DETECTION EXPERIMENTS......61
3.5 FIXED CALIBRATION......64
3.6 FIXED CALIBRATION EXPERIMENTS......74
3.7 SELF-CALIBRATION......81
3.8 SELF-CALIBRATION EXPERIMENTS......85
3.9 CONCLUSIONS......91

4 CORRESPONDENCE ESTIMATION......93

4.1 INTRODUCTION......93
4.2 GEOMETRIC CORRESPONDENCE IN IMAGE PAIRS......96
4.3 CLASSIC CORRESPONDENCE ESTIMATION METHODS......100
4.4 BAYESIAN CORRESPONDENCE ESTIMATION......102

4.5 DENSE-FIELD REPRESENTATIONS 104
4.6 JOINT PROBABILITY MODEL FOR THE FIELDS 111
4.7 CRITERIA FOR BEST SOLUTIONS 119
4.8 SEARCH ALGORITHMS 121
4.9 BAYESIAN DESIGN OF NEW ESTIMATORS 125
4.10 CONCLUSIONS 133

5 3-D SCENE VISUALIZATION ON STEREO DISPLAYS 137

5.1 INTRODUCTION 137
5.2 GENERATION OF MULTI-VIEWPOINT IMAGES 139
5.3 GEOMETRICAL ERRORS 142
5.4 EXPERIMENTS 152
5.5 CONCLUSIONS 159

6 PANORAMA REAL-TIME 3-D VISUAL COMMUNICATION SYSTEM 161

6.1 INTRODUCTION 161
6.2 SCENE MODEL FOR REAL-TIME SYSTEMS 163
6.3 ACQUISITION WITH A STEREO CAMERA 165
6.4 SCENE VISUALIZATION 168
6.5 A TWO-WAY VIDEOCONFERENCING APPLICATION 173
6.6 PANORAMA SYSTEM 177
6.7 EXPERIMENTS 182
6.8 CONCLUSIONS 187

7 FUTURE OUTLOOK ON 3-D VISUAL COMMUNICATION SYSTEMS 189

7.1 CURRENT STATUS OF 3-D SYSTEMS 189
7.2 DIRECTIONS FOR FUTURE RESEARCH 190

A GEOMETRY NOTATION 193

B EPIPOLAR GEOMETRY 201

C GIBBS AND MARKOV RANDOM FIELDS 203

D MARKER DETECTION ALGORITHM 207

BIBLIOGRAPHY 221

SUMMARY 231

SAMENVATTING 235

ACKNOWLEDGEMENTS 239

CURRICULUM VITAE 241

Chapter 1

Introduction

1.1 Human vision and technology

Human vision plays a major role in everyday life. We use it to gather information about the environment around us. If a person is physically close to an object of interest, he can inspect the object by vision. The object emits or reflects light, the light propagates in space, and finally enters the viewer's eyes. The brain then draws conclusions about the presence and state of the object and decides what actions to take, such as "do nothing", "take", "eat" or "run away as fast as you can".

Human vision has its limitations. Whenever the distance between the scene and the viewer is too large, or the scene is too small, the resolution of the eyes is too small to see any details. This can be justified by the fact that small and distant objects are often less important than large and nearby objects. However, the drive to explore the world and improve living standards (and less peaceful reasons) made us develop tools to enhance or assist human vision.

The telescope is the first device for the inspection of distant objects, or in short, television. Although no precise date is known of its invention, lenses forming its main component have been found in Greece dating from about 2000 BC [Ceo]. The telescope requires a long path between viewer and object in which the light can travel freely, without being blocked by some other object. For relatively small distances on the earth's surface, the telescope is a powerful tool. For long-distance television, however, high magnification factors make accurate aiming of the telescope more and more difficult. Objects easily get in between the light path, or the curvature of the earth intervenes. The inspection of our universe seems the only application in which these limitations play a minor role.

The first television system in the usual sense of the word is the machine invented by Carey in 1875 [Ebo], see Figure 1.1. It records an image with a number of light-sensitive devices arranged in a 2-D array. The resulting electric signals are transmitted in parallel and fed into a 2-D array of light bulbs, which visualize the image. This system was simplified substantially by the introduction of the mechanical scanner by Nipkow [Ebo]. A fast rotating disk scans the image linewise and the "pixels" in the image are transmitted sequentially, requiring only one electric channel.

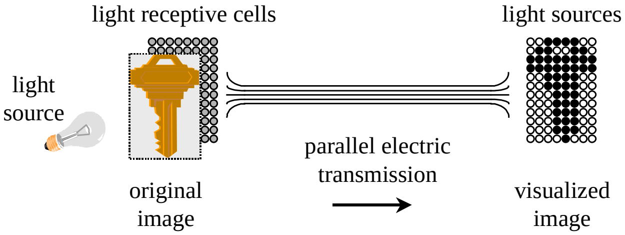


Figure 1.1 The Carey television system.

In both the Carey and Nipkow television systems, three different parts can be distinguished, which are also present in the modern visual communication systems of today:

- Acquisition of the scene
- Transmission in some kind of representation
- Visualization of the scene

Through time, the list of available technology for 2-D image-based visual communications has grown to a substantial size [Davi92]. For scene acquisition, we have many types of Cathode-Ray-Tube based and Charge Coupled Device (CCD) cameras. For scene visualization we have displays such as Cathode Ray Tubes (CRT), Liquid Crystal Displays (LCD) and plasma displays. The representations used are analog (PAL, NTSC, SECAM, HDMAC, MUSE), digital (CCIR601) or efficiently coded digital (MPEG2, MPEG4). To deal with digital signals, we have accurate analog-digital converters, and computers or Digital Signal Processors (DSP), which become more powerful every day. For transmission we have high-bandwidth channels on the basis of cables (coax or glass fiber) and free-air electromagnetic waves (ground and satellite links).

We also would like to store the scene to be able to visualize it at some later time instant. Examples of early systems that have storage capabilities are photography and motion film, which are based on celluloid. More recently video recorders have been developed, which have as a basis tape and magnetic/optical discs.

All this technology allows us to acquire, transmit, store and reconstruct visual information about a scene. At the start, the introduction of long-distance vision was possible only if several concessions were made, such as low-resolution black-and-white still-image photography. Currently the standard of television systems includes high-resolution color video imagery. What is still missing?

1.2 Systems for 3-D visual communication

The ultimate goal in communication can be thought of as the holodeck from the popular science fiction TV series “Star Trek”. Figure 1.2 depicts such an ideal system. Two scenes *A* and *B* are somehow merged into a new virtual scene *C*. The new scene *C* allows for full interaction between *A* and *B*, including sight, hearing, smell, touch and taste.

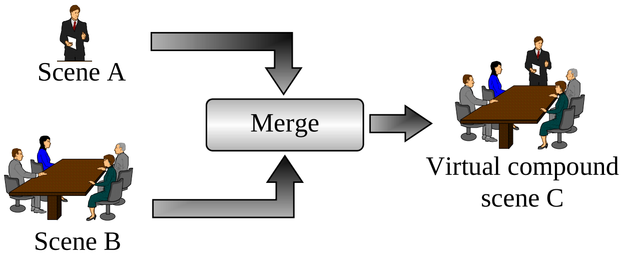


Figure 1.2 The ideal communication system merges two scenes completely. The holodeck from the popular TV series Star Trek is such a system.

The holodeck is still science fiction. At this point, we will drop the more difficult senses touch, smell and taste. Especially for the touch sense, the current state of technology is nowhere near to handling it. The remaining audiovisual communication system is still much more than a television set, as it provides 3-D impressions of the scene. The applications for such a system are obviously enormous. It may enhance the effectiveness of education and interpersonal communication (videoconferencing), enable remote surgery or expert consultancy in the medical areas, provide a means for remote maintenance in hazardous industrial environments, or increase the impact of news or movie broadcasts, entertainment and games, expositions (musea and galleries) and advertising.

We will concentrate on the visual part of the 3-D communication system (much work has already been done to include 3-D sound impressions, e.g. surround processors). A rapidly increasing number of researchers are working on different types of 3-D visual communication systems, such as stereo television systems [Dist95] and holographic systems [Luce97]. In the recently finished European PANORAMA project [Pano98a], a so-called multi-viewpoint system was developed and built. Its properties are more or less in between the stereo and holographic systems, combining the superior quality of holograms with the technological implementability of the stereo system.

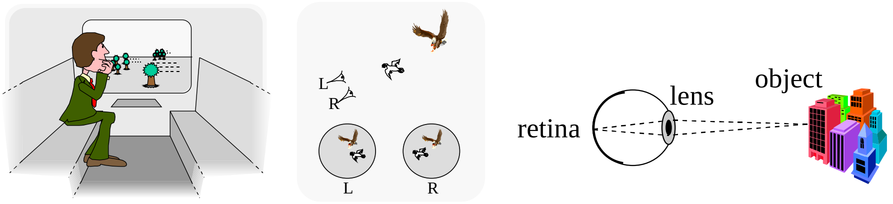
Next, we will examine what to expect from a 3-D visual communication system compared to normal television. Then we review classic, current and future candidates for 3-D systems.

1.2.1 Visual cues to be introduced by 3-D systems

In visual communication, current television systems provide “flat” 2-D images of our world, which clearly has three dimensions. To be able to evaluate the performance of any 3-D visual communication system, let us explore what a viewer is able to observe in the ideal holodeck system, or equivalently, in normal surroundings.

All objects in a scene emit or reflect light rays in different directions, with different intensities and colors. The viewer only observes the light rays that pass through the pupils of his eyes in the direction of the retinas. On the retinas 2-D images of the scene are formed, containing photometric information. To extract geometric and more complex semantic information, the brain uses several cues that can be categorized in psychological and physiological cues [Ebo]. The psychological cues involve knowledge about the scene, such as known shapes of certain objects, the way shade works, occlusion (objects that move in front of each other) relative size between objects, perspective and color. These cues are all provided by the current television systems.

The physiological cues are the absent cues that we would like to introduce by 3-D communication systems, see Figure 1.3. These cues are more related to direct measurable physical properties, such as the accommodation of the eye lens and the convergence of the eyeballs towards an object of interest. They all have something in common: they are related to our "3-D" experience of the scene. We categorize them as follows by scale of viewer position (large), inter-eye distance (medium) and pupil size (small).



(a) **(b)** **(c)**

Figure 1.3 Visual cues to be introduced by 3-D communication systems, ***a)*** motion parallax, ***b)*** stereo, and ***c)*** accommodation.

Viewer position: motion parallax cue

If the viewer walks or moves his head, his viewpoint changes. For a continuously moving viewer, nearby objects seem to move faster than objects that are far away. This is called the motion parallax cue. Figure 1.3a shows a viewer moving in a train. The nearby trees move faster than the distant trees. Besides providing the depth of objects, this cue might be used to assist the balancing organ since it provides direct feedback of the viewer's own motion.

Inter-eye distance: stereo cue

The viewer has two eyes, separated by some moderate distance. This provides him with two slightly different viewpoints, which enable him to determine the distances to the objects. This is called the stereo cue. Figure 1.3b shows the different positions of the two birds on the viewer's retinas, a cue for their difference in distance.

Pupil size: accommodation cue

The pupils of the eyes have finite size and thus, from each point of the object, several light rays forming a cone enter each eye. The eye lens diffracts all these light rays, which then form a cone within the eyeball. If the lens is accommodated to the correct distance to the point of interest, the apex of the cone is at the retina, producing a sharp image of that point, see Figure 1.3c. The required accommodation provides a cue for the distance to the object. This is called the accommodation cue. The cue also provides a means to focus attention on the object of interest, since all objects at other distances become blurred.

All three cues mentioned invoke relatively simple geometric models rather than complex semantics about objects. This fact is comfortable from a technological point of view, since these cues are the ones we would like to integrate into 3-D visual communication systems.

1.2.2 Classic, current and future 3-D systems

Normal television
A normal television set presents the same image to both eyes, and thus lacks the stereo cue. To a certain extent, the motion parallax and accommodation cues are provided. However, they are not related to the movement and position of the viewer, but to those of the camera. Presumably, when looking at a scene at a TV screen, a viewer imagines himself as being physically near the displayed scene and then identifies himself with the camera. However, the actual position and movement by the viewer does not relate in any way to the presented motion parallax cue. The accommodation of the eyes is related only to the distance between the viewer and the display, and thus, independent of the selected point of interest in the scene. Compared to the ideal situation of actually being present in a scene, a television set visualizes a scene that has the geometry of a flat rectangle (the screen) on which rich and dynamic photometric effects are shown.

Stereo systems
The first system that provided the stereo cue was stereo photography [Turi]. It has been quite a success in books, expositions and toys such as the View Master, which is still available (see Figure 1.4). These systems require the viewer to look into two tubes that provide different images in each eye.

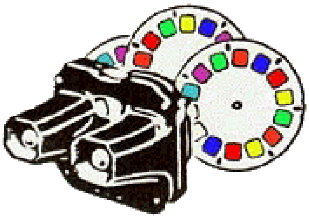


Figure 1.4 The View Master, a system introducing the stereo cue. It provides a sense of depth.

Later on, stereo film and TV were not a large breakthrough, since a severe concession was made to allow multiple viewers at the same time, called anaglyphy. The images for the left and right eye were shown simultaneously on the same display with different colors: one was red, the other cyan. Each of the viewers wore red/cyan glasses and observed the two images in their left and right eyes separately. Unfortunately the completely different colors are a good reason to complain about a headache. Other systems have been devised to overcome this, such as polarization and LCD shutter glasses [Sext99]. The latter was used in the European RACE DISTIMA project [Dist95], where a high-quality and full-color stereo system was built for the first time.

For all stereo television systems, the stereo cue is introduced without motion parallax and lens accommodation. The absence of motion parallax results in geometric distortion of the visualized scene [Rede00]. Figure 1.5 shows that only one viewer can observe the scene correctly. This viewer must remain still at a specific and fixed position relative to the display.

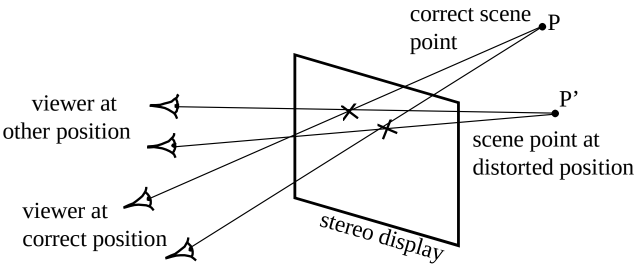


Figure 1.5 Stereo without motion parallax. Only a single viewer at a specific, fixed viewpoint observes the scene correctly. Any other viewpoint results in geometric distortion.

The absence of the accommodation cue results in a conflict between the accommodation and convergence of the eyes [Herm71, Rede00], see Figure 1.6. The eye lenses are accommodated to the display, while the eyes are rotated both to converge on the scene point of interest. This provides the brain with two conflicting depth cues.

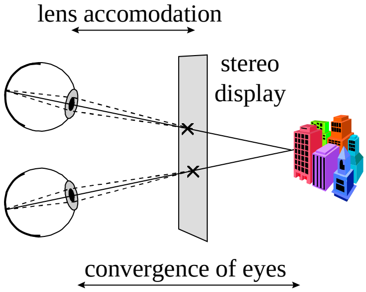


Figure 1.6 Stereo without accommodation produces the accommodation-convergence conflict.

Thus, although a stereo system provides an impression of depth, the brain is not provided with three consistent cues. This might be experienced as being worse than normal television systems, where none of these three cues are present. The design of a stereo setup that minimizes these discomforts for all viewers is a complex task, involving careful setup of cameras, display, and position of the viewers [Dist92, Herm71, Past91, Kutk94, Ariy98].

Fixed multi-viewpoint systems

For the motion parallax cue, several systems have been made. The first is xography, introduced in 1964 for the well-known 3-D postcard [Gisc93], see Figure 1.7. A scene is recorded by a normal camera from several viewpoints. The images are then combined in some special way and printed on the card. A special sheet of lenses is put on top of the card, which gives it a rough surface. When one looks at the card in different directions, two of the original images each reach on of the eyes. Since no glasses are required, this kind of stereoscopy is referred to as autostereoscopy. For a stereo system the number of images is just two, and only the stereo cue is provided. When the number of images exceeds two, the system is a so-called multi-viewpoint system. Movement of the viewer's head equals selection of a pair of images, corresponding to the new viewpoint. Effectively, the motion parallax cue has been added. However, the viewpoints are restricted to a fixed and discrete range.

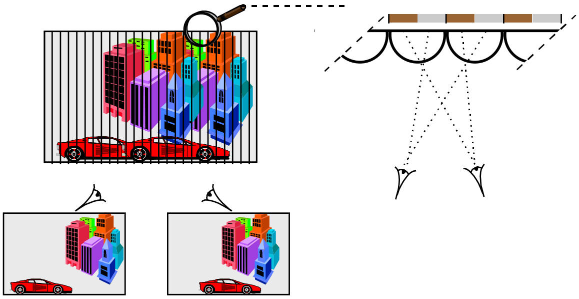


Figure 1.7 The introduction of motion parallax by xography or 3-D postcard. It provides a limited, discrete range of viewpoints.

The same principle can be used in electronic displays by placing a special sheet of lenses in front of the display, a so-called lenticular screen. Currently, these displays are commercially available with about ten images [Phil]. Both the 3-D postcards and the lenticular screen displays are still not perfect. They do not provide the accommodation cue. A special version of a multi-viewpoint display is described in [Kaji97], with a number of viewpoints as high as 45. If the allowed range of viewer motion is as low as 10 cm, the resolution of the motion parallax cue is then about 2 mm, enabling even multiple viewpoints within each pupil of the viewer. It was reported that this effect produces the accommodation cue. The 10 cm viewing range minus the inter-eye distance provides only a few centimeters of motion parallax. Effectively, the accommodation cue was introduced at the cost of the motion parallax cue.

The drawback of all these approaches is that only a discrete number of viewpoints is available, which have been selected during the acquisition of the scene.

Adaptive multi-viewpoint systems

A different type of multi-viewpoint system provides only two images at the same time, but adapts these to the current position of the viewer. This restricts the number of viewers to one, but the motion parallax can be continuous in any range. Recently, such a system has been built for the first time in the European Community sponsored PANORAMA project [Pano98a], see Figure 1.8. The real-time adaptation of the images requires a large amount of digital processing. In the project, this was realized by dedicated hardware that could perform stereo image interpolation. An autostereoscopic display was used, that adapts its lenticular screen to the current positions of the viewer's eyes. The viewer did not need to wear glasses.

The adaptive multi-viewpoint system is able to produce the stereo and continuous motion parallax cues. Depending on future developments in autostereoscopic displays, the accommodation cue may be incorporated. This requires at least two views per pupil, so four in total. To handle more than one person, the number of views goes up linearly with the number of persons.

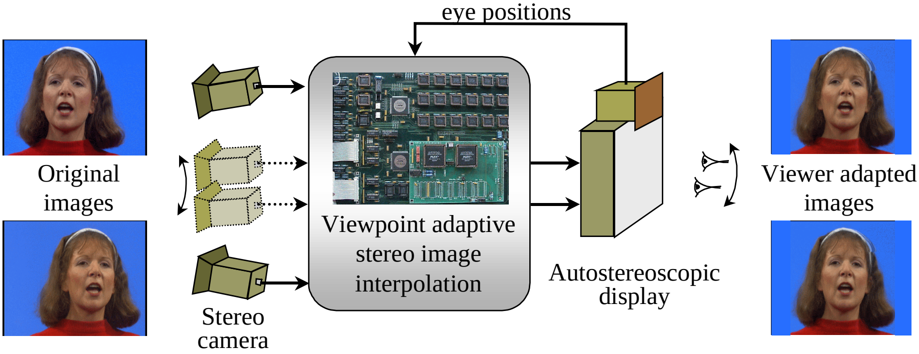


Figure 1.8 Continuous motion parallax produced by the PANORAMA system via image interpolation. It requires state-of-the-art real-time signal processing, implemented in hardware.

Holographic systems

For the future, it seems that hologram technology is the most promising candidate for 3-D visual communication systems: an infinite number of viewpoints in a continuous range are provided simultaneously. This enables all three cues for multiple viewers without glasses. In a hologram, a photograph with a very high-resolution is used to record so-called fringe patterns, or interference patterns of light that is reflected by the scene. The patterns contain all information necessary to reconstruct the scene visually for any viewpoint. The patterns have the scale of the wavelength of visible light, which is about 500 nm. A hologram of any reasonable size thus contains a tremendous amount of information. At this point, technology only allows high-resolution full-color holography for still images. Prototype real-time systems, so-called holovideo systems, have been built with concessions to size, resolution and color of the scene [Luce97], still requiring a massive parallel supercomputer to process all data.

Overview

Table 1.1 gives an overview of the systems mentioned and their characteristics. All these systems are so-called “through-the-window” systems, see Figure 1.9. They use a planar display to visualize a 3-D scene to the viewer, which allows the viewer to observe the scene as if looking through a window (the display) [Sext99]. The display can only reconstruct light rays that pass through it. Clearly, walking around to the back of the display does not provide us with a look at the back of the scene, thus limiting the motion parallax capability.

The display and the current positions of the viewer’s eyes define a pyramid-shaped visibility region in which the virtual scene must reside. When the viewer moves, the region moves. This forms a restriction on the scenes that can be visualized, or equivalently, on the position and gaze direction of the viewer.

System	Technology	# viewers with correct scene visualization	Cues provided Stereo	Motion parallax	Accomo-dation
Photography / TV	classic current	0	-	-	-
Stereo	classic current	1	√	-	-
Fixed multi-viewpoint	current	1..N	√	restricted	future
Adaptive multi-viewpoint	current	1	√	√	future
Holography	future	∞	√	√	√

Table 1.1 Systems for 3-D visual communication.

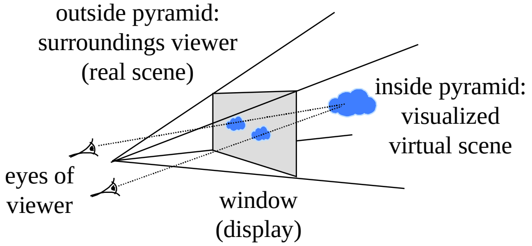


Figure 1.9 A through-the-window based 3-D display system.

For example, a very large scene can only be visualized if it is positioned far behind the display, where the pyramid is wide enough to contain it. Then, the viewer cannot change his viewpoint, since in that case the pyramid would move away from the scene. This leads to a severely reduced motion parallax possibility. If the system reacts by repositioning the scene into the visibility pyramid, the viewer will notice that the scene ‘follows’ his movements, which is very unnatural. In stereo systems without viewpoint adaptation, this is what effectively happens (in Figure 1.5, the scene point automatically follows the visibility region).

Several solutions are available to overcome the limitations of the window-based display systems. First, we can enhance these systems by providing the viewer with a means to reposition and scale the visualized scene manually [Rede00]. Walking around the scene to see the back can be simulated by manual rotation of the scene. By appropriate scaling and translation, the scene can be visualized centered in the display. This gives the viewer almost

180° of motion parallax, while the virtual scene remains in the visibility region. In addition, this minimizes the accommodation-convergence conflict. This is at the cost of the scale correctness of the scene visualization.

Another solution is provided by a different system, called the immersive 3-D system. It completely surrounds the viewer, or his eyes, to allow for all positions and gaze directions. The system comes in two different sizes. In the CAVE concept [Cruz93], the viewer can walk freely within a cube of 3x3x3 meters. On four or five of its faces, stereo images are displayed, continuously adapted to the viewer's position. Thus, it is based on four or five adaptive multi-viewpoint systems working simultaneously. A much smaller immersive system only covers the eyes of the viewer by means of a Head Mounted Display [Hmd]. In this case two adaptive multi-viewpoint systems are fixed to the viewer's head, each of them providing only a single monoscopic image.

1.3 Scope of the thesis

In this thesis, we will consider a "through-the-window" based adaptive multi-viewpoint system. Figure 1.10 shows this system dotted, in the context of all previously mentioned communication systems. We will concentrate on the specific implementation depicted in Figure 1.11, which was built in the PANORAMA project. The main application will be 3-D videoconferencing.

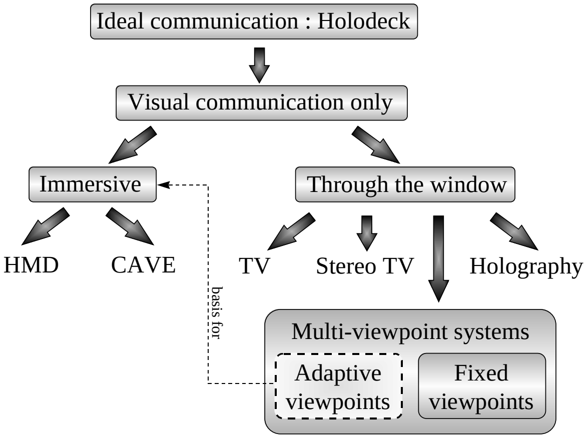


Figure 1.10 Overview and context of 3-D visual communication systems.

The PANORAMA system is based on stereo equipment. Scene acquisition is performed by a stereo camera and scene visualization by a stereo display. In between, a 3-D scene model is transmitted. Signal processing is needed to transform the stereo data into a 3-D model (analysis) or vice versa (synthesis). In the synthesis part, the position of the viewer's eyes has to be measured to allow for viewpoint-adaptive scene visualizations.

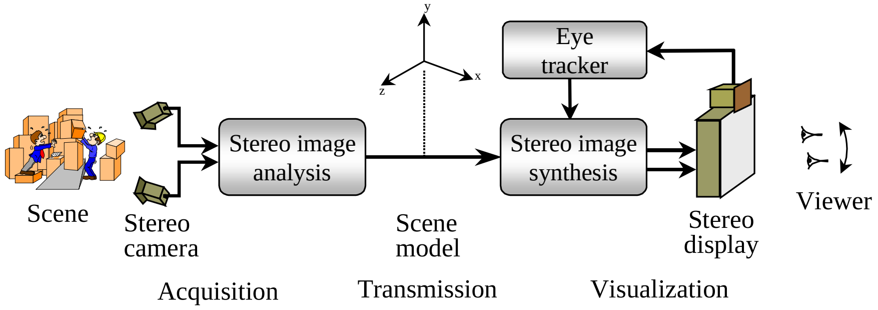


Figure 1.11 The PANORAMA adaptive multi-viewpoint system. It employs scene acquisition with a stereo camera and scene visualization with a stereo display.

We will deal with three tasks: stereo image analysis, stereo image synthesis and the overall PANORAMA system design including the choice of the scene model. We do not consider transmission, efficient coding or storage of the scene model. Further, we use commercial available cameras, (auto-) stereoscopic displays and eye trackers [Orig, Phil, Ster]. Next we will elaborate on the stereo image analysis, stereo image synthesis and system design tasks.

1.3.1 Acquisition of a 3-D scene with a stereo camera

The principle of acquiring 3-D information with a stereo camera is shown in Figure 1.12. The internals of two cameras are shown, comprising lenses and image planes. A correspondence field describes a pixel-dense set of pixel pairs in the stereo image pair. Each such pair of pixels is lit by the same scene point by two light rays. If the two light rays are constructed backwards, from the pixels through the lenses towards the scene, their intersection yields the 3-D coordinates of the scene point. This process is called triangulation. It requires that the position and orientation of the two cameras (lens and image planes) are known. Together these parameters form a geometrical model for the stereo camera.

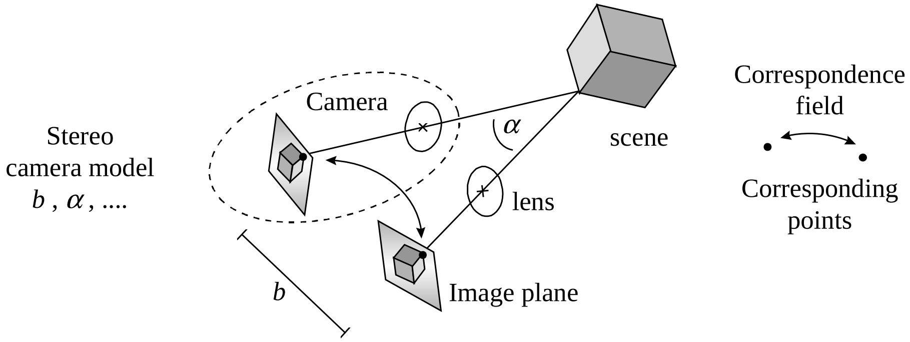


Figure 1.12 The acquisition of a 3-D scene by a stereo camera requires two tasks. By correspondence estimation, pixel pairs are found that originate from the same scene point. By camera calibration, a geometrical model for the cameras (position and orientation of image planes and lenses) is found. Together, for each pixel pair two light rays can be reconstructed. Their intersection yields the 3-D coordinates of a scene point.

The acquisition of the camera parameters and the correspondence field is done by camera calibration and correspondence estimation respectively. We elaborate on these aspects next.

Stereo camera calibration
Camera calibration emerged first in photogrammetry [Slam80] and has been studied for many years [Wolt78, Tsai87, Faug93, Pede99]. It can be performed before recording the actual scene by so-called fixed calibration. In this method, a special object with an accurately known geometry and photometry is recorded by the cameras. By detecting special points of the object (markers) in the images, and using the object's known geometry, one can calibrate the cameras accurately. The drawback of this method is that it requires user interaction and the accurate (expensive) manufacturing of the calibration object. Further, each time a change is made in the camera setup, such as zooming in or out, the calibration has to be repeated

Current research is devoted more and more to self-calibration, in which no user interaction and no object are needed [Arms96, Boug98, Eela99, Faug92, Rede98d]. Self-calibration is performed on the basis of a correspondence field, obtained from an image pair of the scene after correspondence estimation. The absence of accurate geometric knowledge about the scene makes it harder to estimate the camera parameters. First, since we have no reference to the standard SI meter, the absolute scale of both the camera geometry and the scene cannot be obtained. Secondly, it can be proven that at most seven parameters can be obtained if the cameras have ideal lenses [Faug93]. Any stereo camera model that has more parameters results in ambiguities. So, for successful calibration we need extra constraints or knowledge [Deve96], such as a known pixel aspect ratio [Poll98].

We will investigate fixed and self-calibration algorithms. Our main goal will be to unify these two approaches. We will design them both in a similar and flexible fashion using the Bayesian probability framework. We will examine lens distortion in detail and show that its incorporation in self-calibration enables us to estimate more than the aforementioned 7 parameters. This increases the applicability of self-calibration.

Correspondence estimation
For correspondence estimation, a tremendous amount of algorithms can be found in literature. They range from feature matching [Barn80, Liu93], block matching [Acca95, Haan92, Kana94, Hend96], pel-recursive [Biem87, Börö91] and optical flow techniques [Horn86, Enke88, Tsai97]. In the past years, more and more algorithms are designed using the Bayesian probability framework [Drie92, Chan94, Heit93, Konr92, Teka95b, Stil97, Woo96] on the basis of Markov Random Field models, introduced in image processing in [Gema84].

Many of these algorithms perform so-called motion estimation, defined for objects in two frames from an image sequence from a single camera. Correspondence estimation algorithms for stereo images from calibrated cameras are called disparity estimation algorithms [Cox96, Inti94, Fran96, Ohta85, Rede98a]. It can be shown that for stereo images, all correspondences must lie on lines in the images, the so-called epipolar lines [Faug93, Truc98]. The position of these lines is only known if the cameras are calibrated. In this case, the search space for the correspondences has a one-dimensional nature along these lines.

For stereo images from uncalibrated cameras, the position of the epipolar lines is not known in advance. Therefore we have to resort to correspondence algorithms that search in a 2-D way all over the images, similar to motion estimation algorithms. This severely increases the computational complexity. In such cases, a three-step approach is often used [Poll98], in which first a low number of accurate correspondences is estimated involving a two-dimensional search with pre-defined features such as corners. The feature correspondences are then used for self-calibration of the cameras. After calibration, the position of the epipolar lines is known and the more efficient (one-dimensional) disparity estimation can be performed to obtain a pixel-dense correspondence field.

We will derive new correspondence estimators for stereo images from both calibrated and uncalibrated cameras. For this we first set up our requirements for the estimator. We will review current correspondence estimation algorithms thoroughly and select the Bayesian framework as the ideal tool for our estimators. A detailed examination of the steps in this framework will lead us to several new findings that are included in our estimators. A major goal is the development of an estimator that combines high quality results with low computational requirements. Further we require the estimator for uncalibrated cameras to be robust for rotational and scale differences between the images. This is required for self-calibration of a camera pair with unequal orientation and zoom factors.

1.3.2 Visualization of a 3-D scene on a stereo display

For the viewpoint-adaptive visualization of a 3-D scene on a stereo display, many image synthesis algorithms can be found in literature. However, mostly image formation is considered for the images from virtual cameras [Faug96, Fuji96, Levo96], often restricted to viewpoint interpolation of the original stereo images [Chup94, Fran96, Rede97d, Seit95, Tsen95, Veig96]. It is not clear whether the images formed by these algorithms can truly visualize a 3-D scene on a display without geometrical errors. Further, no analyses have been reported on other sources of visualization errors, such as eye-tracking errors.

We will derive the correct algorithm for geometrically correct scene visualization. It will turn out that the algorithm is very similar to, but not the same as, image formation from a virtual camera. Further we analyze the error in the 3-D scene visualization caused by eye-tracking errors and rendering latency.

1.3.3 Overall design of the PANORAMA system

As the PANORAMA system was the first multi-viewpoint system actually built, little can be found about the design of similar systems. Most designs consider stereo systems [Ariy92, Dist95, Grin94, Herm71, Kutk94, Past91].

In this thesis, we will describe all parts of the PANORAMA system. In specific, we will pay much attention to the choice of scene model used for transmission. As this choice influences both the analysis and synthesis parts of the system, it has a large impact on the system complexity. Then we will examine the overall system settings for optimum performance. We will show that even with all the implementation constraints, a system can still be built with geometrically correct scene visualization.

In the system description, we will make use of theoretical results derived earlier for the different system parts. However, the PANORAMA system does not use all algorithms derived in this thesis, for reasons of implementation feasibility and mainly the fact that the system design was fixed in 1995 for project continuity.

1.4 Thesis outline

The stereo image analysis part of the thesis consists of Chapter 2 (stereo camera models), Chapter 3 (stereo camera calibration) and Chapter 4 (correspondence estimation). Chapter 5 deals with the stereo image synthesis part. Chapter 6 deals with the PANORAMA overall system design. Chapter 7 concludes the thesis with an outlook on the future of 3-D visual communication systems.

In Chapter 2, models for stereo cameras are investigated. We start by introducing a new notation, based on tensor notation from physics. This will help us to denote the many-parameter models clearly. Further, it simplifies the review of many models from literature and enables to pinpoint their similarities and differences. We describe a general stereo camera model and examine various reductions to less complex models. One of those is the parallel camera setup, which is used in the PANORAMA system. The triangulation process is described, needed to acquire the 3-D scene points after camera calibration and correspondence estimation have been performed. Finally, simple rules of thumb are derived to decide on the camera setup for recording a scene with specific dimensions.

In Chapter 3, fixed and self-calibration of stereo cameras will be investigated. A new, accurate and very robust algorithm is derived for detecting special markers in images. We will design a fixed calibration algorithm using the Bayesian probability framework, Simulated Annealing (SA) algorithms and the marker detection algorithm. The self-calibration algorithm is designed in a similar way using a correspondence field instead of markers. All algorithms are extensively tested on both synthetic and natural images.

In Chapter 4, we will examine correspondence estimation algorithms. After a brief review of the classic methods, we will focus on the modern Bayesian methods that estimate pixel-dense fields with possibly sub-pixel accuracy. The review is based on [Rede99a]. In this framework, we will present two new estimators, one for calibrated cameras and one for uncalibrated cameras. The latter provides the input for self-calibration algorithms. It must be able to cope with any stereo camera setup, e.g. any position and orientation of the two cameras. Both algorithms use Markov Random Field (MRF) models and Simulated Annealing (SA) minimization algorithms. The combination of MRF/SA with the often used hierarchical estimation approach has the potential of providing high quality correspondence fields, while keeping the computational load reasonable opposed to most current MRF/SA algorithms. Experiments are performed to validate the quality and computational load of the algorithms. The quality of the correspondence fields is examined both in terms of 3-D scene model quality and robustness for camera setups (large differences between the two cameras in e.g. orientation).

In Chapter 5, we will derive an image synthesis algorithm for geometrically correct scene visualization on stereo displays. It will turn out that the algorithm is very similar to but not the same as image formation in a normal camera. Further we analyze the error in the 3-D scene visualization caused by eye-tracking errors and system latency. This analysis is based on [Rede00]. All theoretical results will be validated by extensive subjective experiments, using real-time computer simulations with synthetic images.

Chapter 6 deals with integration of all components into the PANORAMA 3-D videoconferencing system. The results are based on [Ohm98, Pano98b, Pano98c, Rede97a, Rede97b, Rede97c, Rede97d, Rede97f, Rede00]. First we examine several options for scene models and make a choice that enables a feasible system implementation. We describe the cameras, eye tracker, stereo display and image analysis/synthesis parts. For a feasible implementation, several choices and concessions were made. These include carefully placing the cameras into some prescribed setup (making calibration unnecessary) and using image interpolation for view generation. We will show that in spite of all these restrictions, a 3-D visual communication system can still be built that provides geometrically correct scene visualization. We design one-way and two-way communication systems and show that the two-way system is more complex than the sum of two one-way systems. Extensive experiments will be performed, both with computer simulations and the actual PANORAMA system. It will be shown that the introduction of motion parallax by adaptive multi-viewpoint systems indeed enhances the feeling of telepresence compared to normal or stereo television. Finally, we examine the directions for future research in the area of multi-viewpoint systems.

Chapter 7 concludes the thesis with an outlook on the future of 3-D visual communication systems.

Chapter 2

Stereo camera models

2.1 Introduction

In this chapter we will introduce a geometric model for stereo cameras. The geometric model relates 2-D image coordinates with light rays in 3-D space. In our application, 3-D scene acquisition with a stereo camera, corresponding light rays are triangulated to yield 3-D scene points. The triangulation process needs a geometric model for the specific stereo camera with which the images of the scene are recorded. We will discuss a general, parameterized stereo camera model, the accompanying triangulation process and the resulting general properties of the acquired scene, which include position, size, accuracy and resolution. Figure 2.1 illustrates the scene acquisition process: the parts enclosed by dotted lines are treated in this chapter.

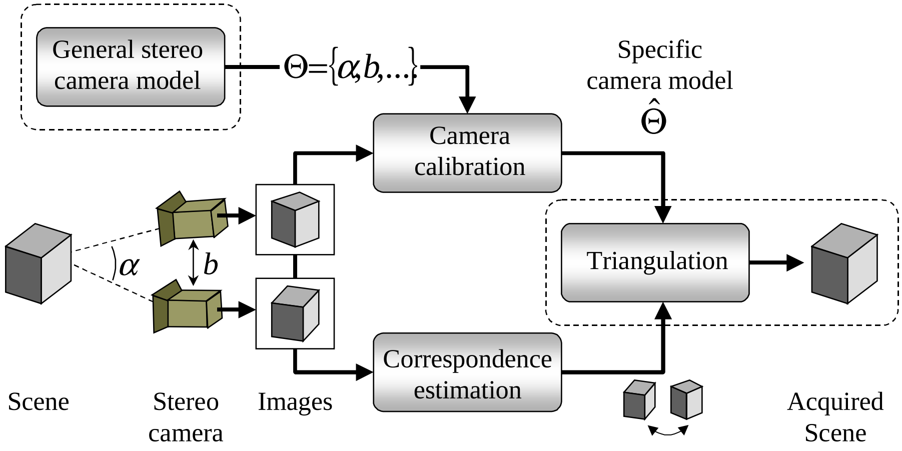


Figure 2.1 The scene acquisition process. This chapter deals with the general stereo camera model, the triangulation procedure and the properties of the acquired scene (shown dotted).

The general camera model contains all possible stereo camera setups. By camera calibration (Chapter 3) we estimate the model parameters. The estimated parameters determine which of all possible setups is the specific model for the actual cameras, which is used by the triangulation process. The estimated specific camera model and the estimated correspondence field (Chapter 4) determine the properties of the acquired scene.

A huge amount of literature is already devoted to the modeling (and calibration) of cameras [Faug93, Luon93, Pede99, Roth97, Truc98, Tsai87, Wei94]. This has led to several different models and different notations. Especially the models differ among different calibration methods. In fixed calibration methods, where a special calibration object is used, most approaches use complex models, which describe many properties and artifacts of cameras (mispositioning of the CCD chip, lens distortion). The model parameters are usually related to directly measurable lengths and angles, giving rise to so-called explicit models [Roth97,Wei94]. In self-calibration, where the scene itself is used as calibration object, many approaches use the so-called fundamental matrix as a model [Faug93, Long81, Truc98]. This matrix captures only a limited part of the stereo camera geometry in a single 3x3 matrix. The entries do not refer directly to measurable physical quantities, and thus the matrix constitutes an implicit model. Scenes cannot be acquired with exact geometry using this model (e.g. scale and angles may not be correct). The model is often used because it has been proven at most 7 parameters can be measured by self-calibration [Arms96, Csur97, Luon93]. Thus an explicit model with more parameters can be designed, but is useless since not all of its parameters can be calibrated. However, the 7-parameter proof relies on the condition of ideal pinhole cameras, that is, without lens distortion. For practical cameras with lens distortion, there is no such proof. Yet, lens distortion is never modeled in self-calibration in an attempt to avoid the 7-parameter limit.

The main goal of our camera model review is to unify most current camera models, on the level of notation, model properties and the ability to be estimated by both fixed and self-calibration methods. We accomplish this by using a tensor notation, which is used in physics [Schu85]. We choose to keep the models as explicit or physically possible, since this makes it easier to understand the camera models and to verify the estimated parameters [Pede99]. Special attention will be given to lens distortion, which possibly enables self-calibration with more than 7 parameters. The general model will be used in Chapter 3 on camera calibration. We will describe several simpler models found in literature, such as pinhole models (without lens distortion) and the parallel stereo camera that plays a major role in chapters 4 and 6 of this thesis. Additionally, we will describe the triangulation process in detail, used in chapters 4 and 6. Finally, we will derive rules of thumb to setup the stereo camera before making a recording to match the intended scene dimensions, resolution and accuracy.

The chapter is organized as follows. The tensor notation is motivated in section 2.2. Then in section 2.3 we will outline our general stereo camera model using this notation, and compare it with other current models. Section 2.4 deals with the simpler camera models. In section 2.5 we will examine the triangulation procedure that corresponds with our camera model. Then in section 2.6 the properties of the acquired scene are determined, as a function of the selected camera setup and the estimated correspondences. The chapter is concluded by section 2.7.

2.2 Geometric notation

In the area of camera modeling, a wealth of different notations is used. The models usually have a number of parameters in the order of ten to twenty. If a corresponding pixel pair and its triangulated scene point are considered, about 30 parameters may come into play. Vector and matrix notations are generally used both to use the alphabet effectively and to group the parameters into meaningful sets. These notations require a very clear definition. Accidental mistakes such as using a matrix instead of its inverse ($Ax=y$ instead of $Ay=x$) are easily made.

We will use a notation based on tensor notation in physics [Schu85]. Our motivation is:

- A clear definition of variables is included in the notation itself (accidental mistakes such as reversal of vectors or inversion of matrices are avoided).
- All parameters, scene and image points/coordinates share a similar notation.
- Parameters from other models are easily translated into this notation.

In a large part of camera calibration literature, so-called projective geometry is used, see e.g. [Arms96, Faug93, Poll98, Roth97, Truc98]. This is a powerful mathematical tool that enables us to denote transformations such as $y=Ax+b$ (linear) and $y=1/x$ (non-linear) very compactly in a single, linear 4x4 matrix transformation $\hat{y} = \hat{A}\hat{x}$. The drawback is that variables are introduced without physical meaning (each 3-D vector is denoted by a 4-D vector, which contains only 3 meaningful numbers) and that it cannot include other non-linearities such as lens distortion. We will not use projective geometry, but our notation can be extended to this notation by just adding an extra coordinate.

The notation is also used in other chapters of this thesis, and can be found in Appendix A.

2.3 A general stereo camera model

In this section we describe our general stereo camera model. It incorporates all current models such as the basic pinhole model, lens distortion and mispositioning, misorientation and skew of the CCD chip. We assume that CCD cameras are used, but the results in this chapter hold also for other types of cameras.

The parameters in the camera model are categorized in external (or extrinsic) and internal (intrinsic) parameters [Faug93, Luon93, Pede99, Truc98, Tsai87]. The external parameters describe the position and orientation of the cameras with respect to their environment, such as α and b in Figure 2.1. The internal parameters describe the properties of the lenses and CCD chips within the cameras, such as the size of the pixels, lens aberrations, and the focal length. The focal length is the distance between lens and CCD, which determines the camera viewing angle or zoom. Focal length is also defined as a property of the lens [Brow71], but if the scene is projected sharply on the CCD, the two definitions are the same.

Figure 2.2 illustrates our model in our notation. It shows the relation between a scene point P_S and its projections onto the left and right CCDs, the points P_{PL} and P_{PR}. The point P_S emits (or reflects) light, of which two rays are modeled, the light rays l_{SL} and l_{SR}, which enter the camera through the centers of the lenses. Due to lens distortion effects, the lenses may refract the rays, which results in the diffracted rays l_{IL} and l_{IR}. The indices *S* and *I* stand for the scene side and the image side of the lens (outside and inside the camera housing).

No less than eight reference frames are introduced: the scene frame *SC*, the stereo camera frame *SF*, the lens frames *LF* (*LFL* and *LFR*), the projection plane frames *PF* (*PFL* and *PFR*), and finally, the CCD or image reference frames *I* (*IL* and *IR*). All reference frames have coordinates in meters, except for the image reference frames, which have coordinates in pixels. Table 2.1 shows the reference frames, their coordinates, units and their position.

Frame	Name	Coordinates	Origin position
SC	Scene frame	x_{SC} y_{SC} z_{SC} [m]	Related to scene
SF	Stereo frame	x_{SF} y_{SF} z_{SF} [m]	Midway of baseline
LFL	Left lens frame	x_{LFL} y_{LFL} z_{LFL} [m]	Left lens optical center
LFR	Right lens frame	x_{LFR} y_{LFR} z_{LFR} [m]	Right lens optical center
PFL	Left projection frame	x_{PFL} y_{PFL} z_{PFL} [m]	Center of left CCD
PFR	Right projection frame	x_{PFR} y_{PFR} z_{PFR} [m]	Center of right CCD
IL	Left image coordinates	x_{IL} y_{IL} [pixel]	Upper left corner left CCD
IR	Right image coordinates	x_{IR} y_{IR} [pixel]	Upper left corner right CCD

Table 2.1 Reference frames in the general stereo camera model.

The position and orientation of the lenses, the external parameters, are completely described by the relation between the stereo frame *SF* and lens frames *LF*. The position and orientation of the CCD with respect to the lenses, the internal parameters, are modeled using the relation between the lens frames *LF* and projection frames *PF*. This includes the focal length (lens-CCD distance, or zoom) and errors in chip placement (chip is not centered on, or not orientated orthogonal to the lens optical axis, as shown quite exaggeratedly in Figure 2.2). The physical size of the CCD chips and the pixels, and properties such as skew are modeled using the projection frames *PF* and image frames *I*.

In section 2.3.1 we will first discuss the most important reference frame, *SC*, often called world frame [Truc98, Zhan93], which is used to describe the 3-D scene points acquired by triangulation. It is the only reference frame that plays a role after the calibration and triangulation have been performed, and thus, has direct impact on the application. After that, we will describe the paths of the light rays going in the physical direction from the scene points towards the CCD chips (opposite to the direction of triangulation in section 2.4). Sections 2.3.2 to 2.3.6 discuss five different steps in the model, each invoking a transformation from some point to another point or from some reference frame to another. Where possible we will deal with only one of the cameras and drop the *L* or *R* subscript. Finally section 2.3.7 provides an overview of all parameters in the general stereo camera model.

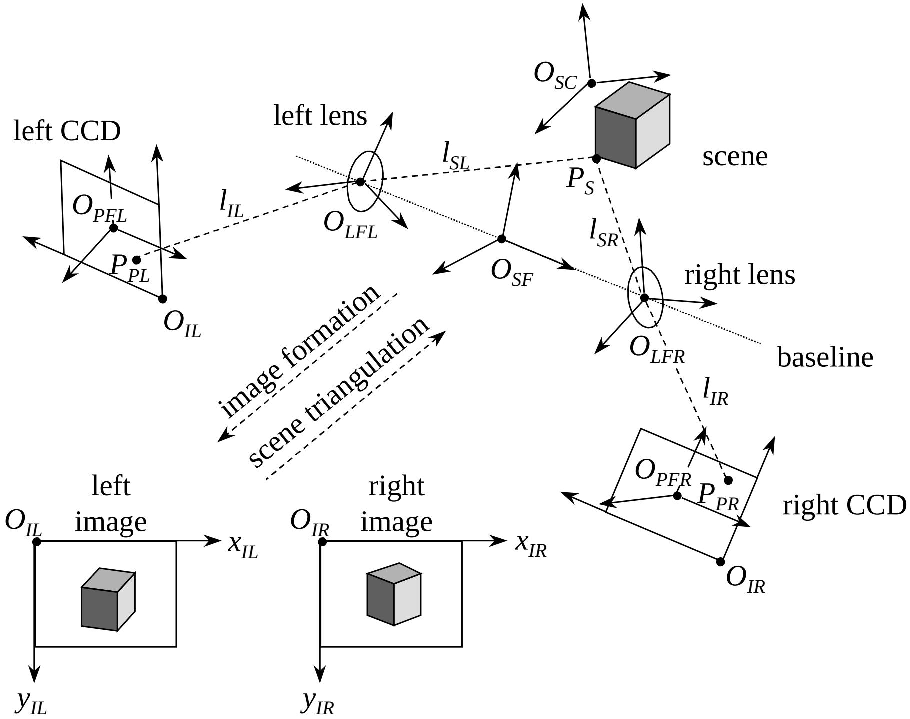


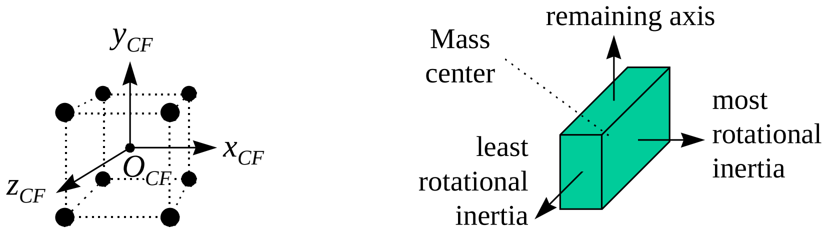
Figure 2.2 The general stereo camera model.

2.3.1 Selection of scene reference frame SC

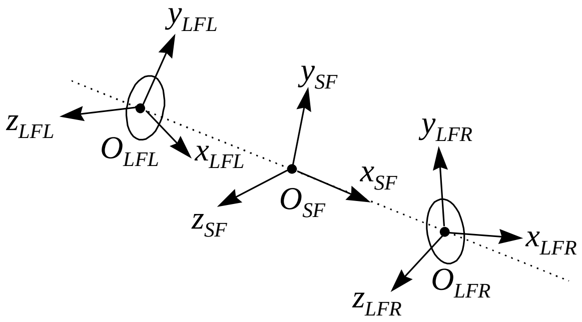
In principle, the position and orientation of the scene reference frame *SC* can be selected arbitrarily. Figure 2.3 shows three ways that are often used to define *SC*. In fixed calibration schemes, we always have a calibration reference frame *CF*, in which the geometric model of the calibration object is given. It can be used directly as the scene frame, thus *SC* = *CF*.

In self-calibration schemes, no *CF* frame is available. Instead we can use the scene itself to define a frame. The center of mass may serve as origin for *SC*, and the principle axes of rotational inertia may define its orientation [Rede99c]. Since these features must be extracted from the acquired scene, we must first define some preliminary scene frame SC^{pre}, acquire the 3-D scene, then extract the features, construct *SC* and finally transform the acquired scene from frame SC^{pre} to *SC*. The selection of SC^{pre} is completely arbitrary. In this scheme, it is not possible to measure movements between scene and cameras.

The third option is to define the scene frame using the cameras. Figure 2.3 gives an example using the optical centers of the two lenses. It yields the *SF* frame that is defined in the center of the two lenses, where the x_{SF} axis points from the left lens to the right lens. This reference frame is used in [Rede98d, Rede99b, Rede99d] and also shown in Figure 2.2. Other options are to use one of lens frames *LF*, e.g. in [Luon93, Zhan93] *LFL* is used.



calibration object actual scene



cameras

Figure 2.3 Definition of reference frame *SC* via a special calibration object, the actual scene or the cameras.

If processing a sequence of stereo images, we must choose a scene frame SC^i for each stereo image pair *i* from the sequence. In fixed calibration approaches, all SC^i are the same, since the calibration is performed only once for the entire sequence. Similarly, in self-calibration of stereo image sequences, we can perform the calibration on the first stereo image pair and use the parameters for the entire sequence. However, if we want to change the cameras (e.g. zoom) during the recording, we have to calibrate the cameras separately for each frame. If we define each SC^i using the actual scene scheme, we may have a problem when the scene deforms over time. The orientation of the *SC* frame can then change discontinuously as a function of the deformation, causing inconsistent rotations of the acquired scene. We can overcome this by introducing temporal consistency constraints in the SC^i frame sequence, at the cost of increased complexity.

We use the *SF* frame as scene reference frame for the following reasons:

- Its definition does not require any scene model
- It does not require any computation (feature extraction from the acquired scene)
- The frame can be used in both fixed and self-calibration schemes.
- In image sequences, movement of the entire scene w.r.t. the cameras can be measured
- It is symmetrical (compared to one of the lens frames)

2.3.2 From the scene frame to the camera frames

The first step for each scene point P_S, given in frame *SF*, is to obtain its coordinates in the cameras' lens frames *LFL* and *LFR*. With (A.15) we get for an arbitrary camera with frame *LF*:

$$P_S^{\sigma_{LF}} = V_{\sigma_{SF}}^{\sigma_{LF}} \left(P_S^{\sigma_{SF}} - O_{LF}^{\sigma_{SF}} \right) \tag{2.1}$$

This equation has six parameters per camera: three for position $O_{LF}^{\sigma_{SC}}$ and three for orientation $\varphi_{SF}^{LF;\sigma}$ (in $V_{\sigma_{SF}}^{\sigma_{LF}}$) that determine the lens frame *LF* with respect to the stereo frame *SF*. These are the external parameters in the model.

We assume that the lenses are perfectly rotationally symmetric around their optical (z_{LF}) axes. Thus, the two angles $\varphi_{SF}^{LFL;z}$ and $\varphi_{SF}^{LFR;z}$ have no meaning. However, we will keep them as valid angles in the model to describe the *z* orientation of the cameras. Later, when the CCD chip is added to the model, we can leave out the *z* angles for the CCD chip with respect to the lens. With this, the model loses some physical clarity but gains a clear distinction in external and internal parameters.

As Figure 2.3 indicates, the position of the *SF* frame is chosen in the center of the two lenses (optical centers). The line through the optical centers is called the baseline. The x_{SF} axis is defined as lying on the baseline, pointing from the left to the right camera. This gives the following:

$$O_{LFL}^{\sigma_{SF}} = \begin{bmatrix} -\tfrac{1}{2}b \\ 0 \\ 0 \end{bmatrix} \qquad O_{LFR}^{\sigma_{SF}} = \begin{bmatrix} +\tfrac{1}{2}b \\ 0 \\ 0 \end{bmatrix} \tag{2.2}$$

The *b* is the distance between the optical centers, which confusingly is also called baseline. The fixation of the x_{SF} axis determines the *SF* frame up to a free rotation around the x_{SF} axis or baseline. We can account for this by requiring that:

$$\varphi_{SF}^{LFL;x} + \varphi_{SF}^{LFR;x} = 0 \tag{2.3}$$

The *SF* frame is now chosen such that it has the average orientation of the two lenses (only with respect to rotation around the x_{LF} axis).

In total, this gives us 6 parameters for the stereo camera. They constitute all external parameters. The *b* determines the position of both cameras on the baseline. The angle $\varphi_{SF}^{LFL;x}$ determines the rotation around the baseline of both cameras (vertical direction of camera viewing zones). The two rotation angles $\varphi_{SF}^{LFL;y}$, $\varphi_{SF}^{LFR;y}$ determine the direction and convergence of the two viewing zones of the cameras. Finally, the two rotation angles

$\varphi_{SF}^{LFL;z}$, $\varphi_{SF}^{LFR;z}$ determine the rotation of the camera around its principal axis (viewing direction).

The fact that the original twelve parameters have been reduced to six is due to the specific selection of the stereo frame *SF* as the scene frame. In other camera models, the external parameters are usually denoted by a translation vector *T* and a rotation matrix *R* for each camera [Azar95, Eela99b, Luon93, Pede97a, Truc98], e.g.:

$$T = O_{LF}^{\sigma_{SC}} \qquad R = V_{\sigma_{SC}}^{\sigma_{LF}} \tag{2.4}$$

In this notation, we must ensure that we are aware which index (or subscript) is up or down. If they are reversed, the vector and matrix have a different meaning.

2.3.3 From 3-D to 2-D

We use the point P_U to model the projective aspect of a camera, that is, its reduction from a 3-D world to a 2-D image. The point P_U is defined on the intersection of the incoming light ray l_U and the plane z_{LF} = -1, see Figure 2.4. This plane is parallel to the lens (orthogonal to the optical axis) and situated one meter behind it at the scene side.

The coordinates of P_U as a function of those of P_S are given by:

$$P_U^{\sigma_{LF}} = -\frac{P_S^{\sigma_{LF}}}{P_S^{z_{LF}}} \tag{2.5}$$

The point P_S has three free coordinates, while the point P_U has only two (the z_{LF} coordinate is fixed to -1).

This step involves no additional camera parameters.

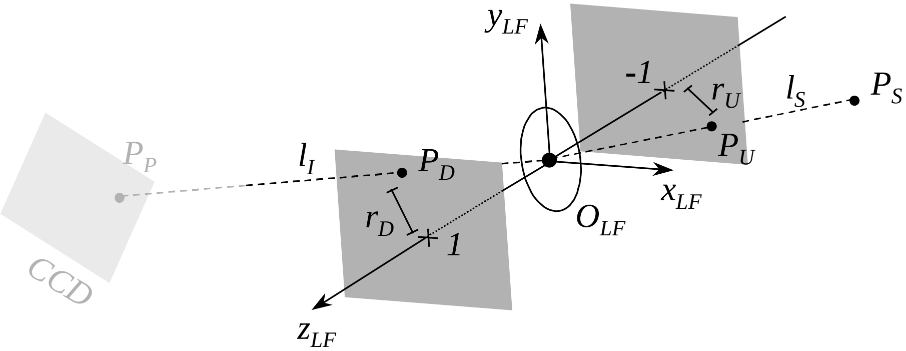


Figure 2.4 The points P_U and P_D in the lens frame are used to model the projective property of the camera and lens distortion.

2.3.4 Lens distortion

All practical lenses suffer from distortions. The projection of a scene point on the image can deviate in the order of 5 pixels from the ideal position [Tsai87], and in rare cases up to 10-100 pixels [Stei97a]. In the triangulation process, this may lead to significant distortion of the acquired scene. Lens distortion can be modeled to a great extent [Slam80], which makes it possible to reduce its effect on the scene acquisition. The additional parameters for the distortion model are internal camera parameters.

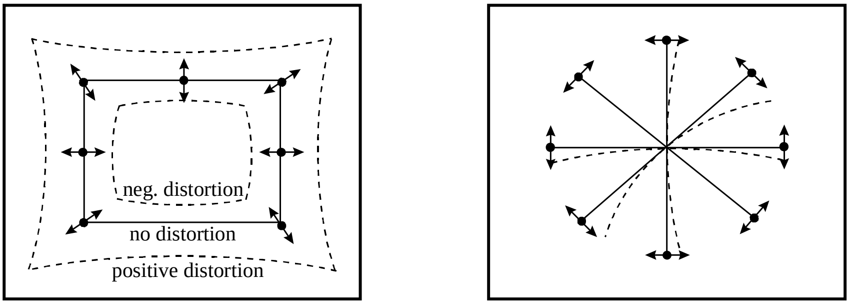
We model lens distortion using the points P_U and P_D. The point P_U is situated on the intersection of the incoming light ray l_S and the plane z_{LF} = -1 on the scene side of the lens, and the point P_D is situated on the intersection of the transmitted light ray l_I and the plane z_{LF} = 1, at the image side of the lens, see Figure 2.4.

Distortionless lens (pinhole camera)
For a distortionless lens, we would find:

$$P_D^{\sigma_{LF}} = -P_U^{\sigma_{LF}} \tag{2.6}$$

Such a model is commonly used [Arms96, Basu95, Csur97, Deve96, Faug92, Faug93, Faug96, Fitz98, Jeba99, Poll98, Truc98, Zhan93, Ziss95]. The simplicity of (2.6) enables an analytic approach to camera calibration at the cost of model accuracy. For these models, projective geometry [Faug93] provides a powerful mathematical analysis tool. Unfortunately, if lens distortion is included, it cannot be denoted in the compact 4x4 projective matrix notation and therefore obscures the analytical advantages of projective geometry.

Types of distortion
Lenses show several types of distortion, of which the effects can be described by so-called radial and tangential lens distortion [Brow71, Eela99a, Heik97, Slam80, Tsai87, Weng92]. Figure 2.5 illustrates these effects in an image of a rectangular and a star-like object. The effects are zero in the image center and largest at the borders.



(a) *(b)*

Figure 2.5 Two types of lens distortion, ***a)*** radial and ***b)*** tangential.

Radial lens distortion is rotationally symmetric around the optical axis of the lens. In rotationally symmetric lenses, this is the only distortion we expect. Tangential distortion arises in practical cameras with multiple lenses, whose optical axes do not coincide exactly [Weng92]. The difference between front and rear surface curvatures of a single lens may produce a similar effect [Heik97]. It has been found by several authors that tangential distortions are negligible and very small compared to the radial distortions [Pede99, Slam80, Tsai87, Wei94]. We will therefore consider only radial lens distortion.

Radial distortion

On the basis of [Slam80], we model radial distortion as follows:

$$\begin{bmatrix} P_D^{x_{LF}} \\ P_D^{y_{LF}} \\ P_D^{z_{LF}} \end{bmatrix} = -\begin{bmatrix} P_U^{x_{LF}} \cdot Q \\ P_U^{y_{LF}} \cdot Q \\ P_U^{z_{LF}} \end{bmatrix} \tag{2.7}$$

The fact that Q is present in only two of the coordinates in (2.7) yields the non-linear distortion. By definition, Q is given by:

$$\begin{aligned} Q &= \frac{r_D}{r_U} \\ r_U^2 &= \left|P_U^{x_{LF}}\right|^2 + \left|P_U^{y_{LF}}\right|^2 \\ r_D^2 &= \left|P_D^{x_{LF}}\right|^2 + \left|P_D^{y_{LF}}\right|^2 \end{aligned} \tag{2.8}$$

The lens distortion model can be parameterized by a series of K_i according to:

$$r_D = r_U + K_3 r_U^3 + K_5 r_U^5 + \ldots.. \qquad Q = 1 + K_3 r_U^2 + K_5 r_U^4 + \ldots. \tag{2.9}$$

In current models, the number of terms is usually one or two. Many different notations are used for the parameters: (K_3,K_5) in this thesis and in [Pede97a, Pede99, Rede98d, Rede99b], (K_1,K_2) in [Stei97a], (κ_1,κ_2) in [Eela99a, Tsai87] and (k_1,k_2) in [Heik97, Truc98, Wei94, Weng92]. We will adopt the two-parameter model as in (2.9). For the stereo camera. This gives four parameters in total.

From Figure 2.4 we can see that the values of both radii r are directly related to the angle of incident and transmitted light rays. For $r = 1$, the ray-optical axis angle is 45°, which corresponds to a ray in a camera with a fairly wide viewing angle (zoomed out). For small-angle cameras (zoomed in), e.g. with a viewing range of -10° to +10°, the radii are in the order of 0.1. To get an impression of the values for the Ks: if the distortion errors are about 5 pixels for practical lenses [Tsai87] and the CCD has about 1000x1000 pixels, then the K_3 and K_5 are in the order of 1 for a small-angle camera ($r \approx 0.1$), and they are about 0.01 for wide-angle cameras ($r \approx 1$).

Figure 2.6 shows two curves of a fairly large radial lens distortion. For some K_3 and K_5, the curve has an extremum (at r_U^{ext} in Figure 2.6). This effect does not appear in any useful real camera.

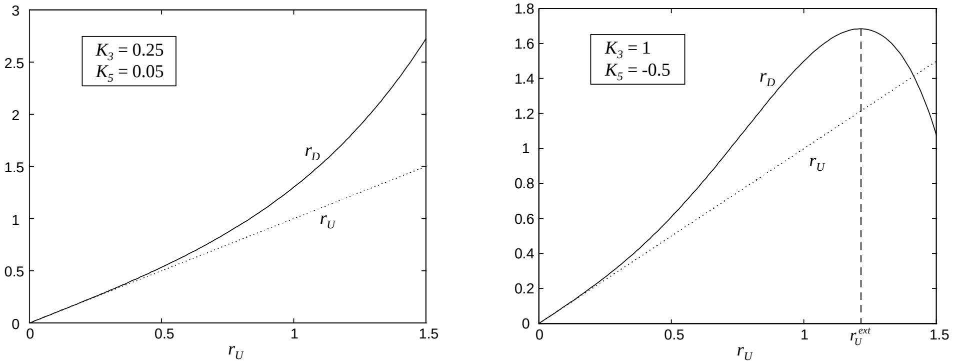


Figure 2.6 Radial distortion curves.

Model coordinates

Most camera models with lens distortion apply (2.7) on image coordinates *I* rather than lens coordinates *LF* [Heik97, Weng92]. This yields other values for the *K*s, which makes it difficult to compare the models numerically [Eela99a].

Our choice of lens coordinates has the following motivation. First, since (2.7) does not require any model of the CCD chip, the parameters are truly lens parameters. In principle, the *K* parameters do not depend on the position of the CCD chip, such as focal length (zoom) and misplacement. For multiple-lens cameras with zoom function, however, the *K*s may however change with zoom. This is not because the *K*s are a function of the focal length, but because both the effective focal length and the *K*s are a function of the complex interaction between the actual lenses. Any change in the *K*s still reflects the fact that the single effective lens really becomes better or worse.

Secondly, if lens distortion is modeled using image coordinates *I*, an equation such as (2.7) results in a more complicated form, as in [Weng92]. All effects of the CCD chip such as pixel size, skew, and CCD mispositioning should then also be taken into account.

Asymmetry in computational efficiency

To compute light rays in both the physical direction and the triangulation direction, we must calculate (2.9) in two different directions (see Figure 2.2). In terms of computational efficiency, equation (2.9) is highly asymmetric.

Since (2.9) is an odd function of r_U, its inverse is also an odd function of r_D:

$$r_U = r_D + K_3^{inv} r_D^3 + K_5^{inv} r_D^5 + K_7^{inv} r_D^7 + K_9^{inv} r_D^9 + ... \tag{2.10}$$

The terms in (2.10) can be computed analytically by replacing each occurrence of r_D with the right-hand side of (2.9) and requiring identity. Even if only two terms are used in (2.9), the series in (2.10) must be infinite if it is to represent the same model. This yields:

$$\begin{aligned} K_3^{inv} &= -K_3 \\ K_5^{inv} &= -K_5 + 3K_3^2 \\ K_7^{inv} &= 8K_3K_5 - 12K_3^3 \\ K_9^{inv} &= K_5^2 - 55K_3^2K_5 + 55K_3^4 \\ K_{11}^{inv} &= \ \ldots\ldots\ldots \end{aligned} \tag{2.11}$$

The coefficients in (2.11) get larger for the higher K^{inv} terms. For practical reasons, the series must be truncated at some point. Then (2.11) is very accurate for small r_D but becomes less and less accurate for larger r_D. A different approach would be to invert (2.9) via interpolation between a number of calculated control points. In [Heik97] a similar method is applied on 2-D image coordinates instead of the 1-D parameter r. These approaches may lead to a higher and more uniformly distributed accuracy with a lower number of terms.

A simpler alternative is to use a numerical method to obtain r_U from r_D. We used the Newton and bisection methods [Pres92]. The Newton method results in:

$$r_{U;i+1} = r_{U;i} - \frac{r_{U;i} + K_3 r_{U;i}^3 + K_5 r_{U;i}^5 - r_D}{1 + 3K_3 r_{U;i}^2 + 5K_5 r_{U;i}^4} \tag{2.12}$$

We start the procedure with $r_{U;0} = r_D$ and proceed until sufficient accuracy has been reached. The bisection method is an iterative procedure that only computes (2.7)-(2.9). It requires an initial interval that must contain the r_U to be computed. We used [0, 2r_D] if the distortion curve has no extremum and [0, r_U^{ext}] if it has one (the extremum can be computed analytically). The final accuracy after N iterations is guaranteed to be within 2^{-N} times the length of the starting interval.

Figure 2.7 shows the computed r_U from r_D as a function of original values of r_U, for two different values of the K_3, K_5 pair. The r_D is calculated with (2.9), and r_U is calculated with the Newton and bisection methods, and with (2.10)-(2.11) using 2, 3 and 19 terms. The analytical method is reasonably accurate for small r but diverges for large r regardless of the number of terms used. For low-distortion cameras (small K), the Newton method reaches an absolute accuracy better than 10^{-12} in about 4 iterations. For wide-angle, high-distortion cameras (large r and large K) 10 iterations are sufficient to reach the same accuracy, but the method does not always converge if the distortion function has an extremum.

We will adopt the bisection method that is simple, applicable for all r and K, reasonably efficient and guaranteed to reach any arbitrary accuracy.

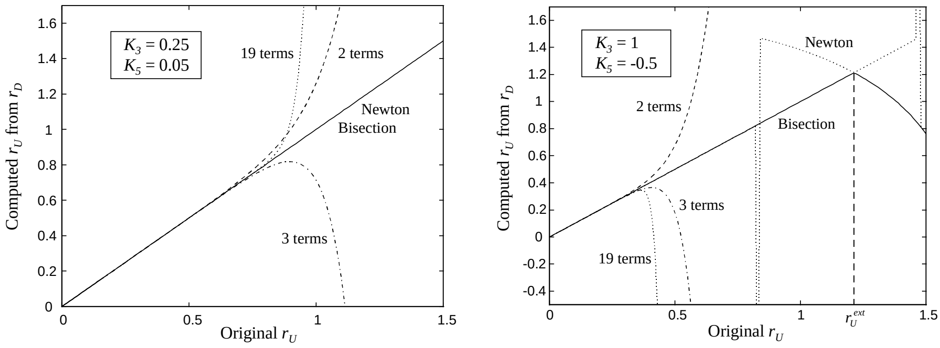


Figure 2.7 Inexact computation of r_U from r_D with analytical and numerical methods.

Asymmetry in modeling

In terms of modeling, (2.9) is also asymmetric. Since a lens is still a lens when turned around (scene side becomes image side and vice versa), we conclude that changing the roles of r_U and r_D in (2.9) yields a different model for lens distortion which has the same modeling quality:

$$r_U = r_D + K_3^* r_D^3 + K_5^* r_D^5 \tag{2.13}$$

The K^*s are new modeling parameters and not equal to K or K^{inv} in the previous paragraphs.

This offers us two different distortion models, (2.9) and (2.13), with equal modeling qualities, but very different computational properties. Whenever an algorithm reconstructs light rays more often in the physical direction from scene towards the images, lens distortion model (2.9) is more attractive. This model is adopted in [Heik97, Rede99b]. If rays are more often reconstructed in the triangulation direction, (2.13) is more efficient. This model is used in [Eela99, Pede99, Stei97, Truc98, Tsai87, Wei94, Weng92].

In our fixed calibration scheme, only rays in the physical direction are used. In our self-calibration scheme, both directions are used in the same proportions. Thus, we adopt model (2.9).

2.3.5 Image formation

The next step is to form an image on the plane in which the CCD chip resides. To this end, we introduce the *PF* frame for each camera. It is defined using only the CCD chip. The relations between the *LF* and *PF* frames will introduce 12 new parameters (6 for each camera). Two will have no meaning, the other ten are directly related to parameters from other models. They model the focal lengths of the cameras and mispositioning and misorientation of the CCD chips.

We assume a perfectly flat CCD, and define that the $z_{PF} = 0$ plane of the projection frame *PF* contains this chip. The optical axis of the lens z_{LF} intersects the $z_{PF} = 0$ plane at the so-called principal point $P_{principal}$ [Arms96, Truc98]. If the CCD chip is positioned correctly, this point lies in the center of the chip. We define O_{PF} to be exactly in the center of the chip and model chip mispositioning via the relation between *LF* and *PF* frames. This relieves us from introducing the point $P_{principal}$ in the model. The final requirement that the x_{PF} axis is parallel to the x_I axis of the CCD chip completely determines the *PF* frame.

In this step we calculate the coordinates of point P_P in the projection reference frame *PF*, see Figure 2.8. The *PF* coordinates are continuous and in meters (we deal with pixel coordinates *I* in the next section).

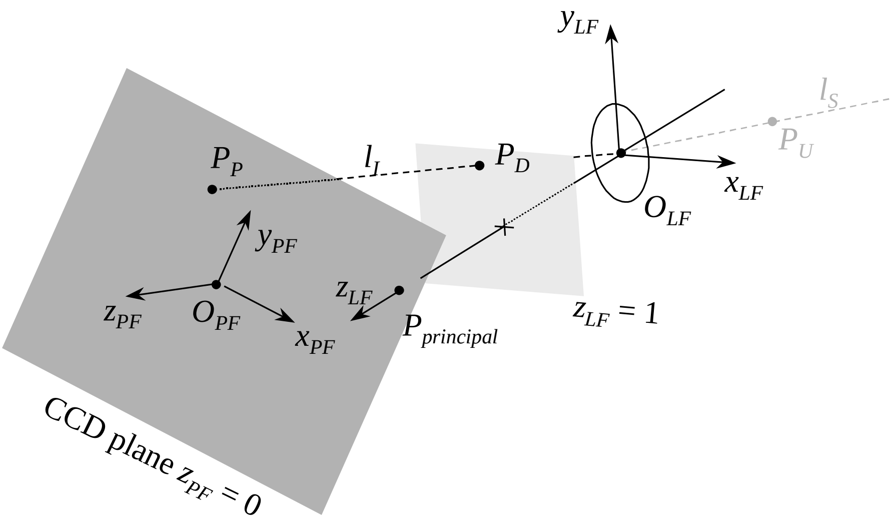


Figure 2.8 Image formation on the projection plane (plane of the CCD chip).

The P_P point lies on light ray l_I, which is most easily described in the lens frame *LF*:

$$P_P^{\sigma_{LF}} = \lambda P_D^{\sigma_{LF}} \qquad \lambda > 0 \tag{2.14}$$

The point P_P also lies on the projection plane or CCD chip, and thus:

$$P_P^{z_{PF}} = 0 \tag{2.15}$$

This leads to:

$$\lambda = \frac{V^{z_{PF}}_{O_{LF}\text{ to }O_{PF}}}{V^{z_{PF}}_{O_{LF}\text{ to }P_D}} = \frac{O_{LF}^{z_{PF}}}{O_{LF}^{z_{PF}} - P_D^{z_{PF}}} \tag{2.16}$$

The *PF* coordinates of P_D (needed in the previous equation) and P_P (the outcome of this paragraph) are:

$$P_D^{\sigma_{PF}} = V_{\sigma_{LF}}^{\sigma_{PF}} \left(P_D^{\sigma_{LF}} - O_{PF}^{\sigma_{LF}} \right) \qquad P_P^{\sigma_{PF}} = V_{\sigma_{LF}}^{\sigma_{PF}} \left(\lambda P_D^{\sigma_{LF}} - O_{PF}^{\sigma_{LF}} \right) \tag{2.17}$$

Equation (2.17) introduces six parameters per camera, which determine the projection frames *PFL* and *PFR* with respect to the lens frames *LFL* and *LFR*, respectively. Since the projection is defined such that the CCD chip resides in the $z_{PF} = 0$ plane and is centered in the x_{PF} and y_{PF} coordinates, the three parameters $O_{PF}^{\sigma_{LF}}$ determine the position of the CCD chip center, measured in *LF* coordinates. The $O_{PF}^{z_{LF}}$ equals the focal length of the camera, that is, the distance between the lens and the CCD chip. It is usually denoted by *f* [Faug96, Heik97, Heba99, Pede97a, Pede99, Truc98, Tsai87, Weng92]:

$$f = O_{PF}^{z_{PF}} \tag{2.18}$$

In the *f* notation we must clearly define what *f* means, in case of chip mispositioning. Then we can define *f* as the distance from O_{LF} to $P_{principal}$, but also as the distance from O_{PF} towards the $z_{LF} = 0$ plane along the z_{PF} axis. Further, the focal length has also been denoted by α [Arms96, Faug92, Zhan93]. Sometimes it is split into two different focal lengths along the vertical and horizontal directions to include the pixel aspect ratio [Arms96, Basu95, Luon93, Poll98].

The other two parameters, $O_{PF}^{x_{LF}}$ and $O_{PF}^{y_{LF}}$, determine the position of the principal point $P_{principal}$ on the CCD chip. They are both zero if the intersection is in the center of the CCD chip. In [Tsai87] practical values for chip mispositioning are found in the order of 10 pixels. Figure 2.8 shows an exaggerated example of CCD mispositioning.

In literature, chip mispositioning is always denoted by the *I* coordinates of the principal point:

$$\begin{bmatrix} u \\ v \end{bmatrix} = P_{principal}^{x_I, y_I} \tag{2.19}$$

Sometimes these coordinates are relative to the CCD chip center (and thus zero in case of zero mispositioning). The coordinates of the principal point have the largest variety of notations encountered in camera calibration: (u,v) in [Poll98], (u_0,v_0) in [Arms96, Faug92, Heik97], (r_0,c_0) in [Weng92], (δ_x,δ_y) in [Basu95], (c_x,c_y) in [Pede97a, Tsai87], (x_{clo},y_{clo}) in [Papa95], (x_0,y_0) in [Pede99], (c_{xr},c_{yr}) in [Stei97a] and (o_x,o_y) in [Truc98]. These notations all need the introduction of the point $P_{principal}$ and need all other aspects of pixel discretization (skew, pixel aspect ratio) in order to denote chip mispositioning. Expressing chip mispositioning in the unit of meters in our notation seems counterintuitive, but in section 2.4 we will see that this will be circumvented in a natural way.

For the orientation of the projection frame (or CCD chip), we have three parameters per camera $\varphi_{LF}^{PF;\sigma}$ in $V_{\sigma_{LF}}^{\sigma_{PF}}$. The $\varphi_{LF}^{PF;z}$ models the rotation of the entire camera housing around the z_{LF} axis. We already modeled this in section 2.3.2 for the sake of a clear distinction between internal and external parameters, and thus:

$$\varphi_{LFL}^{PFL;z} = 0 \qquad\qquad \varphi_{LFR}^{PFR;z} = 0 \tag{2.20}$$

The other two parameters, $\varphi_{LF}^{PF;x}$ and $\varphi_{LF}^{PF;y}$, (two for each camera) model non-orthogonal orientation of the CCD chip with respect to the principal axis of the lens. They are both zero if the CCD chip is oriented exactly orthogonal. They are seldomly modeled in other approaches.

This gives us a total of ten internal parameters for the stereo model. Two model intentional aspects (two focal lengths along the z_F axes) and eight model imperfections (mispositioning/misorientation in the x_F/y_F directions for the left/right CCD chips).

2.3.6 Pixel arrangement on the CCD

The last step is to take into account the pixel arrangement on the CCD, for which purpose we need the pixel size, the pixel aspect ratio and the skew of the CCD chip. Figure 2.9 shows the details of the relation between the *PF* and *I* frames. The image (pixel) coordinates are real numbers, to allow for sub-pixel measurements. The O_I origin is located at the top-left corner of the top-left pixel (right side of Figure 2.9). Therefore, the x_I and y_I coordinates are integers plus one half at the pixel centers.

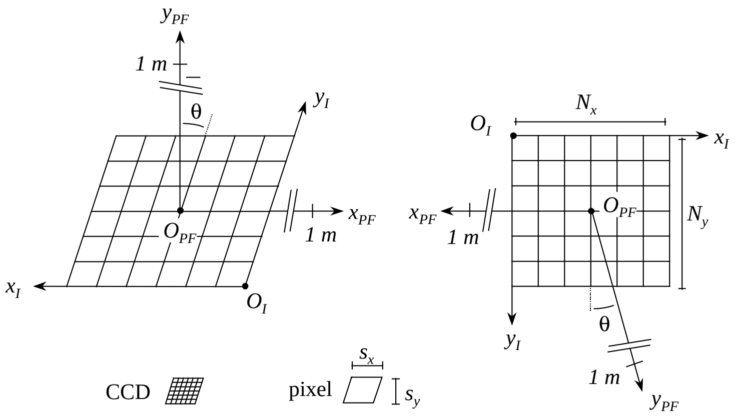


Figure 2.9 Pixel arrangement by the CCD chip. It incorporates skew, scale and translation between projection plane coordinates *PF* (meters) and image coordinates *I* (pixels).

First we calculate the pixel coordinates of P_P, given by the *I* reference frame:

$$P_P^{\sigma_I} = V_{\sigma_{PF}}^{\sigma_I} P_P^{\sigma_{PF}} + O_{PF}^{\sigma_I} \tag{2.21}$$

Since the CCD chip is centered in the *PF* frame, we have:

$$O_{PF}^{\sigma_I} = \begin{bmatrix} \frac{1}{2} N_x \\ \frac{1}{2} N_y \\ 0 \end{bmatrix} \tag{2.22}$$

where N_x and N_y are the horizontal and vertical size of the CCD chip in pixels. These numbers are always known in advance, e.g. $N_x = 720$ and $N_y = 576$ for cameras that yield CCIR601 images.

The base matrix models the size of the pixels and their possibly skewed orientation:

$$V_{\sigma_I}^{\sigma_{PF}} = \begin{bmatrix} -s_x & s_x \tan\theta & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.23}$$

Here s_x and s_y are the horizontal and vertical pixel sizes and θ is the angle with which the y_I axis is skewed (and thus the x_I coordinate, see section 2.2.7). Inversion of (2.23) yields:

$$V_{\sigma_{PF}}^{\sigma_I} = \begin{bmatrix} -s_x^{-1} & s_y^{-1} \tan\theta & 0 \\ 0 & s_y^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.24}$$

which can be used in (2.21). For the stereo camera, this gives a total of six internal parameters: three in (2.23) for each camera.

If models include skew at all, they usually use the θ angle parameter [Arms96, Faug92, Jeba99, Luon93, Zhan93]. Now and then a fractional pixel length unit is used, e.g. $s = s_y^{-1} \tan\theta$ in [Poll98]. The pixel sizes are usually not denoted freely, but combined with the focal length (section 2.3.5) into two compound parameters. This will be treated in section 2.4.1.

2.3.7 Overview of parameters

If all parameters in the sections 2.3.2 to 2.3.6 are taken together, we have a total of $N_{cam\text{-}model} = 26$ parameters, listed in Table 2.2. We call these parameters the physical parameters, where lengths are in meters and angles in radians. The top rows contain the six external parameters; the bottom rows contain the 20 internal parameters.

Function	Parameters	#
Position (x) and orientation (x) of both cameras	$b \quad \varphi_{SF}^{LFL;x}$	2
Orientation (yz)	$\varphi_{SF}^{LFL;yz} \quad \varphi_{SF}^{LFR;yz}$	4
focal lengths	$O_{PFL}^{z_{LFL}} \quad O_{PFR}^{z_{LFR}}$	2
CCD misorientation	$\varphi_{LFL}^{PFL;xy} \quad \varphi_{LFR}^{PFR;xy}$	4
CCD mispositioning	$O_{PFL}^{xy_{LFL}} \quad O_{PFR}^{xy_{LFR}}$	4
Lens distortion	$K_{3;L}\ K_{5;L} \quad K_{3;R}\ K_{5;R}$	4
Pixel size (xy), CCD skew	$s_{x;L} \quad s_{y;L} \quad \theta_L \quad s_{x;R} \quad s_{y;R} \quad \theta_R$	6

Table 2.2 All physical parameters in the stereo camera model. The top rows contain six external parameters, while the bottom rows contain 20 internal parameters. The total number of camera parameters is $N_{cam\text{-}model} = 26$.

2.4 Less complex models

We will reduce the general stereo camera model in several steps, each time discarding a few parameters because they are useless (horizontal pixel size), can be neglected (CCD misorientation and skew), model undesired effects (lens distortion and CCD mispositioning), or cannot be measured (baseline in case of self-calibration). Finally we discuss two special models, the orthographic and parallel camera models. The orthographic model is included just for completeness, but will not be used further. The parallel model will be used in Chapter 4 and in the PANORAMA system in Chapter 6.

2.4.1 Horizontal pixel size

The horizontal pixel size s_x is present in Table 2.2 to make all parameters physical parameters, that is, measurable by hand and easy to understand. However, as is shown in Figure 2.10, if everything on the image side of the lens is scaled arbitrarily (separately for each camera), the stereo image pair remains unchanged.

Thus, the images are invariant for this scaling and we may select one parameter at will. We choose to set the horizontal pixel size to 1 for both cameras:

$$s_{x;L} = s_{x;R} = 1 \tag{2.25}$$

This influences all other length parameters that have meaning on the image side of the lens; the internal parameters. Effectively, horizontal pixel units (hpu) replace the meter as unit of length. A focal length will then typically have values in the order of 500-5000 hpu. The vertical pixel size will reduce to the pixel aspect ratio s_y/s_x. For the CCD mispositioning parameters, the new unit seems a more intuitive choice than meters.

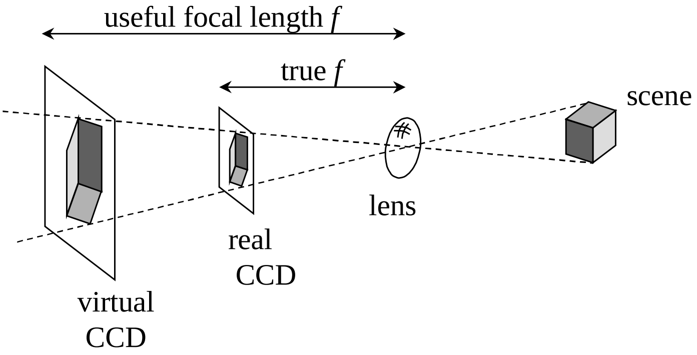


Figure 2.10 Scaling of everything at the image side of the lens has no effect.

Some models from literature use a different reduction and use f/s_x and f/s_y [Arms96, Basu95, Luon93, Poll98], which results in different effective focal lengths in the vertical and horizontal directions. This notation is equivalent to expressing the same focal length with two new and different unit; a concept that is more difficult to grasp than the one we use. Further it is not clear which unit should be used for other internal parameters.

Since triangulation does not require the relation between hpus and meters, we may think of hpus as being equal to meters if this helps us to visualize some parameter values mentally. This results in focal lengths and CCD chip sizes in the order of 1000 meter.

For our application of 3-D scene acquisition we do not need the hpu in meters and use (2.25). This reduces the stereo camera model by two parameters. Table 2.3 shows the new model with 24 useful parameters. The hpu does not need to be the same for the left and right camera (different horizontal pixel length), but this has no consequences.

Function	Parameters	#
Position (*x*) and orientation (*x*), both cameras	b [m] $\varphi_{SF}^{LFL;x}$	2
Orientation (*yz*)	$\varphi_{SF}^{LFL;yz}$ $\varphi_{SF}^{LFR;yz}$	4
focal lengths	$O_{PFL}^{z_{LFL}}$ $O_{PFR}^{z_{LFR}}$ [hpu]	2
CCD misorientation	$\varphi_{LFL}^{PFL;xy}$ $\varphi_{LFR}^{PFR;xy}$	4
CCD mispositioning	$O_{PFL}^{xy_{LFL}}$ $O_{PFR}^{xy_{LFR}}$ [hpu]	4
Lens distortion	$K_{3;L}$ $K_{5;L}$, $K_{3;R}$ $K_{5;R}$	4
Pixel aspect ratio, CCD skew	$s_{y;L}$ $s_{y;R}$ [hpu] , θ_L θ_R	4

Table 2.3 Useful parameters in the stereo model ($N_{cam\text{-}model} = 24$). All internal lengths are now in hpu, while the baseline, the only external length parameter, is still in m.

2.4.2 Vertical pixel size or aspect ratio

Often the vertical pixel size s_y is either not modeled, or equivalently, modeled as 1 (hpu) [Arms96, Azar95, Boug98, Pede97a, Poll98]. This means that the pixel aspect ratio is 1 and the pixels are square. This reduces the stereo model by two parameters:

$$s_{y;L} = 1 \qquad s_{y;R} = 1 \tag{2.26}$$

Many CCD cameras do not have square pixels, but rectangular pixels with aspect ratios of about 0.9 to 1.1. The square pixel reduction is mostly applied in camera models for self-calibration methods to get the number of parameters in the model below the critical value of the 7 that can be estimated with lens-distortion free methods. We will keep the pixel aspect ratio as parameter in our model.

2.4.3 Skew

For modern CCD chips, the skew angle θ is often very small, in the order of 10^{-6} rad [Zhan93], and can thus safely be neglected. This reduces the stereo model by two parameters:

$$\theta_L = 0 \qquad \theta_R = 0 \tag{2.27}$$

2.4.4 Misorientation of the CCD chip

Figure 2.11 shows that CCD misorientation has an effect similar to image warping. The ray l_I hits the misoriented CCD at the edge, but it should have hit a well-oriented CCD q pixels closer to the center. These effects are similar to a combination of radial and tangential lens distortion.

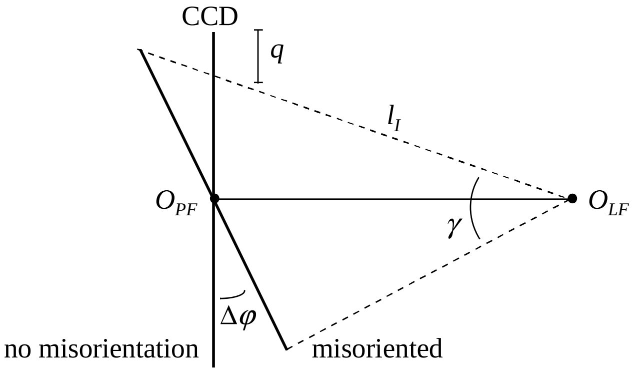


Figure 2.11 CCD misorientation $\Delta\varphi$ described by pixel distortion q. The CCD is viewed in the direction of the misorientation axis.

The pixel errors are largest at the edges of the CCD and zero in the center. We find for the maximum errors:

$$q \approx \frac{\gamma^2}{4}\Delta\varphi \qquad (2.28)$$

For a misorientation angle $\Delta\varphi$ of 1° and a camera with a wide viewing angle of $\gamma = 90°$, we find $q \approx 0.01$, about one percent of a pixel. This number is even lower for points closer to the CCD center, and drops quadratically with viewing angle. Therefore, we can safely discard CCD misorientation from the camera model:

$$\varphi_{LFL}^{PFL;\sigma} = 0 \qquad \varphi_{LFR}^{PFR;\sigma} = 0 \qquad (2.29)$$

2.4.5 Mispositioning of the CCD

The mispositioning of the CCD chip is similar to a rotation of the entire camera, a CCD misorientation and a small change in focal length, shown in Figure 2.12.

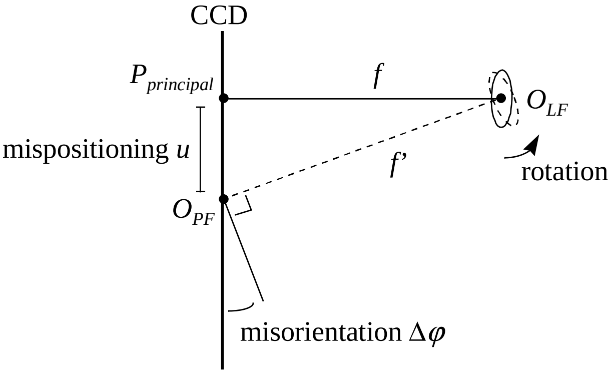


Figure 2.12 CCD mispositioning described by a camera rotation and CCD misorientation.

Table 2.4 shows the two equivalent situations. The amount of mispositioning u is the distance between CCD center and principal point in pixels.

	Mispositioning	Misorientation
CCD mispositioning	u	0
CCD misorientation	0	$\Delta\varphi$
focal length	f	F'
additional rotation	0	$\Delta\varphi$

Table 2.4 The same situation described by CCD mispositioning versus CCD misorientation.

For the $\Delta\varphi$ and f' we find:

$$\Delta\varphi = \frac{u}{f} \qquad f' \approx f\left(1 + \tfrac{1}{2}\Delta\varphi^2\right) \tag{2.30}$$

For u = 25 (very large mispositioning) and f = 500 [hpu] (small for chip with width $N \approx 1000$ pixels) we obtain $\Delta\varphi$ = 0.04 rad ≈ 2°. Via (2.28) we then find (with a large viewing angle γ = 90°) that the remaining distortion in the image is in the order of $q \approx 0.02$. Similar to misorientation, we can thus neglect mispositioning:

$$O_{PFL}^{xy_{LFL}} = 0 \qquad O_{PFR}^{xy_{LFR}} = 0 \tag{2.31}$$

In [Boug98] a similar result is derived. In [Tsai87] and [Zhan93] it is found experimentally that CCD mispositioning has no measurable effect on the scene acquisition.

It thus seems that CCD mispositioning can be safely discarded from the camera model. In all approaches where no lens distortion is modeled, this is the case. If lens distortion is modeled, however, the center of the radial distortion should be at the principle point and not in O_{PF} where it is when mispositioning is discarded. Therefore we may need to model CCD mispositioning to ensure that the model of the lens distortion is correct [Wei94].

2.4.6 Ideal pinhole lenses

If we set K_3 and K_5 to zero for both cameras, we model the lenses to be ideal. The camera model then becomes a so-called pinhole model, which is widely used in calibration [Arms96, Basu95, Csur97, Deve96, Faug92, Faug93, Faug96, Fitz98, Jeba99, Poll98, Truc98, Zhan93, Ziss95]. The name is derived from a sheet of paper with a small (pin)hole in it, which can be used as an ideal lens. The amount of light that is transported by the pinhole is much smaller than by any practical lens, which renders the pinhole often useless in practice. But theoretically, as long as the hole is larger than the wavelength of light, the pinhole has no lens distortion.

As the lens distortion may warp pixel coordinates by about 5 pixels for typical cameras [Tsai87], or even more [Stei97a], these errors cannot be neglected. However, a reason for not modeling them is that the non-linear lens distortion formula obscures analytical treatments of camera calibration. This results in:

$$K_{3;L} = 0 \qquad K_{5;L} = 0 \qquad K_{3;R} = 0 \qquad K_{5;R} = 0 \tag{2.32}$$

Next to losing the distortion parameters, the camera model undergoes an additional change due to the ideal lenses. Since now each lens has the geometry of a point (the hole in the paper), it has no orientation whatsoever. Thus, the lenses also have no specific optical axis anymore. Any line through the optical center may be thought of as an optical axis. Two of these axes have a special role for the CCD chip: one results in zero CCD mispositioning and the other one in zero misorientation, see Figure 2.13. In case of ideal CCD chip placement, both axes are the same.

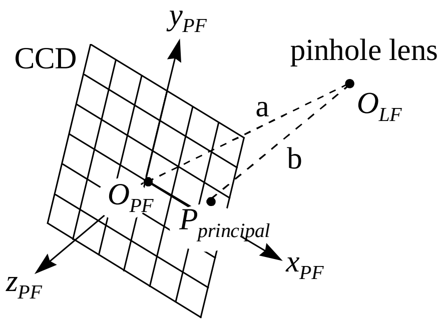


Figure 2.13 A pinhole lens has no specific optical axis. In combination with the CCD chip, two special axes exist, which result in a) zero mispositioning and b) zero misorientation of the chip with respect to the chosen axis.

As we have seen in section 2.4.4, CCD misorientation could be safely discarded from the camera model. We thus select axis *a* as principal axis, which gives zero mispositioning:

$$O_{PFL}^{xy_{LFL}} = 0 \qquad O_{PFR}^{xy_{LFR}} = 0 \tag{2.33}$$

and then safely neglect the misorientation:

$$\varphi_{LFL}^{PFL;\sigma} = 0 \qquad \varphi_{LFR}^{PFR;\sigma} = 0 \tag{2.34}$$

2.4.7 Baseline

In self-calibration approaches, the scene itself is used as calibration object. In general, this scene can be anything and we have no clue about its geometry beforehand. Therefore we have no information about specific lengths in meters of any object. As a consequence, it is not possible to measure any parameter in meters. The only parameter left in the model that needs this unit, is the baseline. In self-calibration approaches we use:

$$b = 1 \tag{2.35}$$

Effectively, the baseline is measured in baseline units (bu). As a result, the acquired scene will also have coordinates in baseline units. With that, we lose the absolute scale of the captured scene. The reduction has no additional effects on the camera model.

2.4.8 Orthographic model

A special kind of camera model is the orthographic model. Whenever the depth size ΔZ of the scene is small compared to the distance Z to the camera, in the order of 5% or lower [Truc98], perspective effects are very small. An example of such a situation is when an object is viewed from a large distance with very small-angle cameras (large zoom or focal length). In this situation, the camera model undergoes severe changes and becomes orthographic [Truc98]; also referred to as a weak-perspective camera [Arms96]. Several parameters from our model then merge into compound parameters or can no longer be defined as before. We will not use this model.

2.4.9 Parallel camera model

The parallel camera model is a very simple model with very few parameters. Figure 2.14 shows the model. Usually for this setup, the CCD chips are drawn at the front of the pinhole instead of behind it. The model assumes an ideal pinhole lens and ideal CCD placement within both cameras, and identical focal lengths, pixel size and orientations among the two cameras. The name of the setup is due to the fact that the CCDs are aligned in parallel (and similarly, the optical axes perpendicular to the CCDs).

The remaining parameters are the baseline *b*, focal length *f* and pixel size s_x, s_y. In combination with the horizontal pixel size, pixel aspect ratio and baseline reductions in sections 2.4.1, 2.4.2 and 2.4.6 respectively, only the focal length parameter remains.

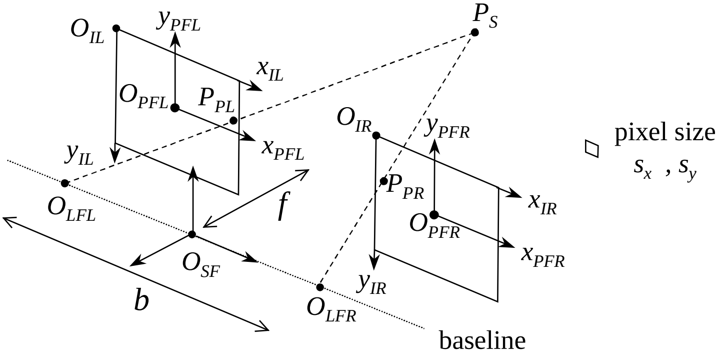


Figure 2.14 The parallel camera setup.

The parallel model is not very attractive for modeling of actual cameras, unless a stereo camera is used with very high quality lenses and sufficient mechanical stability to be aligned manually into the parallel setup. The popularity for the parallel camera model is due to the following. Whenever a stereo camera is calibrated, its images can be warped in such a way that they appear as if recorded by a virtual parallel stereo camera. This warping procedure is called image rectification [Papa95, Roy97].

The parallel setup is attractive because many image-processing tasks are simpler with images from such cameras. For the modeling of light rays in the direction from a scene point P_S towards the left and right images, it follows after some calculation that:

$$\begin{bmatrix} P_{PL}^{x_{IL}} \\ P_{PR}^{x_{IR}} \\ P_{PL}^{y_{IL}}, P_{PR}^{y_{IR}} \end{bmatrix} = \begin{bmatrix} \tfrac{1}{2}N_x \\ \tfrac{1}{2}N_x \\ \tfrac{1}{2}N_y \end{bmatrix} - \frac{f}{P_S^{z_{SF}}} \begin{bmatrix} \left(P_S^{x_{SF}} + \tfrac{1}{2}b\right)/s_x \\ \left(P_S^{x_{SF}} - \tfrac{1}{2}b\right)/s_x \\ -P_S^{y_{SF}}/s_y \end{bmatrix} \tag{2.36}$$

which is much simpler than the entire scheme of section 2.3. The projections have the properties:

$$\begin{aligned} P_{PL}^{x_{IL}} &\geq P_{PR}^{x_{IR}} \\ P_{PL}^{y_{IL}} &= P_{PR}^{y_{IR}} \end{aligned} \tag{2.37}$$

The fact that the *y* coordinates are the same in the left and right image lowers the complexity of correspondence estimation algorithms significantly. All corresponding pixels lie on the same scan line, requiring only a one-dimensional search algorithm. This will be used in chapters 4 and 6.

The triangulation procedure is similarly simple, see section 2.5.5.

2.5 Triangulation

Our application, the acquisition of 3-D scenes, needs triangulation of two light rays that originate from corresponding pixels in the left and right camera images (see Figure 1.12). This requires the construction of two light rays from the CCD chips, through the lenses into 3-D space. In section 2.3, we constructed light rays from 3-D space towards the CCD chips. Here we will execute the same steps in reverse order.

Assume that P_{PL} and P_{PR} form a corresponding pixel pair, of which the coordinates are known in the *IL* and *IR* image frames respectively (via correspondence estimation). Our goal is to obtain the *SF* coordinates of P_S, which now plays the role of acquired scene point.

This section uses the general model with all the parameters that were introduced in section 2.3. The upper and lower subscripts are interchanged whenever it promotes readability and understandability. The treatment does not involve any new parameters.

2.5.1 From the pixel grid back to the projection plane

In section 2.3.6 we calculated the *I* coordinates of a point P_P given its *PF* coordinates by (2.21). Its inversion gives (see Appendix A):

$$P_P^{\sigma_{PF}} = V_{\sigma_I}^{\sigma_{PF}} P_P^{\sigma_I} + O_I^{\sigma_{PF}} \tag{2.38}$$

2.5.2 From the projection plane towards the lens

In section 2.3.5 we determined the position of point P_P in the *PF* frame on the basis of point P_D in the *LF* frame. The reverse is somewhat simpler. First we calculate:

$$P_P^{\sigma_{LF}} = V_{\sigma_{PF}}^{\sigma_{LF}} P_P^{\sigma_{PF}} + O_{PF}^{\sigma_{LF}} \tag{2.39}$$

and then:

$$P_D^{\sigma_{LF}} = \frac{P_P^{\sigma_{LF}}}{P_P^{z_{LF}}} \tag{2.40}$$

2.5.3 Lens distortion

To invert lens distortion, we first calculate r_D with (2.8), use the bisection method to obtain r_U and find Q with (2.8). For P_U it then follows from (2.7) that:

$$\begin{bmatrix} P_U^{x_{LF}} \\ P_U^{y_{LF}} \\ P_U^{z_{LF}} \end{bmatrix} = -\begin{bmatrix} P_D^{x_{LF}} / Q \\ P_D^{y_{LF}} / Q \\ P_D^{z_{LF}} \end{bmatrix} \tag{2.41}$$

2.5.4 Triangulation from stereo 2-D to 3-D

First we calculate the *SF* coordinates of the point P_U :

$$P_U^{\sigma_{SF}} = V_{\sigma_{LF}}^{\sigma_{SF}} P_U^{\sigma_{LF}} + O_{LF}^{\sigma_{SF}} \tag{2.42}$$

After performing all previous steps for both the left and the right camera, we have two points P_{UL} and P_{UR} and their coordinates in the *SF* frame. Figure 2.15 shows that we are now ready to perform the triangulation using the positions of the left and right optical centers (lens frames).

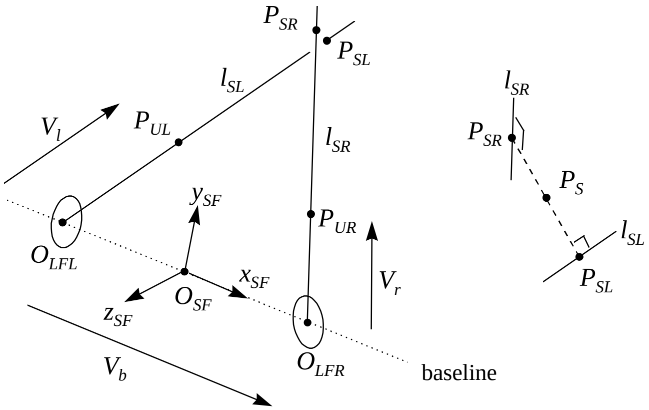


Figure 2.15 Acquisition of a scene point P_S by triangulation of P_{UL} and P_{UR}.

If we denote:

$$\begin{aligned} V_b &= V_{O_{LFL} \text{ to } O_{LFR}} \\ V_l &= V_{O_{LFL} \text{ to } P_{UL}} \\ V_r &= V_{O_{LFR} \text{ to } P_{UR}} \end{aligned} \tag{2.43}$$

then we observe that

$$\begin{aligned} P_{SL} &= O_{LFL} + \lambda_L V_l \qquad && \lambda_L > 0 \\ P_{SR} &= O_{LFR} + \lambda_R V_r \qquad && \lambda_R > 0 \end{aligned} \tag{2.44}$$

with λ_L and λ_R such that

$$\frac{\partial}{\partial \lambda_L} \left| V_{P_{SL} \text{ to } P_{SR}} \right| = 0 \qquad \frac{\partial}{\partial \lambda_R} \left| V_{P_{SL} \text{ to } P_{SR}} \right| = 0 \tag{2.45}$$

and $|V|$ denotes Euclidean distance or length in the *SF* frame (see section 2.2.5). This yields:

$$\begin{bmatrix} \lambda_L \\ \lambda_R \end{bmatrix} = \frac{1}{|V_l|^2 |V_r|^2 - |V_l \cdot V_r|^2} \begin{bmatrix} |V_r|^2 & V_l \cdot V_r \\ V_l \cdot V_r & |V_l|^2 \end{bmatrix} \begin{bmatrix} V_b \cdot V_l \\ -V_b \cdot V_r \end{bmatrix} \tag{2.46}$$

where the dot products and lengths are defined by the *SF* frame (see Appendix A). Then, with (2.44) we obtain the *SF* coordinates of P_{SL} and P_{SR}. Ideally, these points are the same. If the camera parameters or the original corresponding pixel pair contain errors, the points are not exactly the same. Therefore, we construct P_S by taking the average of the points P_{SL} and P_{SR}:

$$P_S^{\sigma_{SF}} = \frac{1}{2}\left(P_{SL}^{\sigma_{SF}} + P_{SR}^{\sigma_{SF}} \right) \tag{2.47}$$

We call the vector $V^{\sigma_{SF}}_{P_{SL} \text{ to } P_{SR}}$ between the two points the intersection error:

$$V^{\sigma_{SF}}_{P_{SL} \text{ to } P_{SR}} = P_{SR}^{\sigma_{SF}} - P_{SL}^{\sigma_{SF}} \tag{2.48}$$

This is a measure of the correctness of the correspondences and the camera model parameters.

2.5.5 Triangulation with the parallel camera setup

In case the parallel camera setup is applicable, the triangulation procedure is much simpler. The intersection error is always zero in this case. From (2.36) we can directly determine the inverse operation:

$$\begin{bmatrix} P_S^{x_{SF}} \\ P_S^{y_{SF}} \\ P_S^{z_{SF}} \end{bmatrix} = \frac{b}{\left(P_{PL}^{x_{IL}} - P_{PR}^{x_{IR}}\right)s_x} \begin{bmatrix} \frac{1}{2}\left(P_{PL}^{x_{IL}} + P_{PR}^{x_{IR}} - N_x\right)s_x \\ -\frac{1}{2}\left(P_{PL}^{y_{IL}} + P_{PR}^{y_{IR}} - N_y\right)s_y \\ -f \end{bmatrix} \tag{2.49}$$

According to (2.37), the two y_I coordinates are the same, which gives some freedom in the selection of the middle equation in (2.49).

2.6 Properties of the acquired scene

In this section we will determine the position, size, accuracy and resolution of the acquired scene. They are determined by the camera model described in section 2.3 (and 2.5), and the accuracies of the estimated camera parameters (see Chapter 3) and the estimated stereo image correspondences (see Chapter 4). Because a large number of parameters are involved in combination with non-linearities due to perspective projections and lens distortions, such an analysis is very complex and hard to find in literature. In [Tsai87] a bound is derived for the accuracy of the acquired scene, but it is quite complex and does not provide a quick insight.

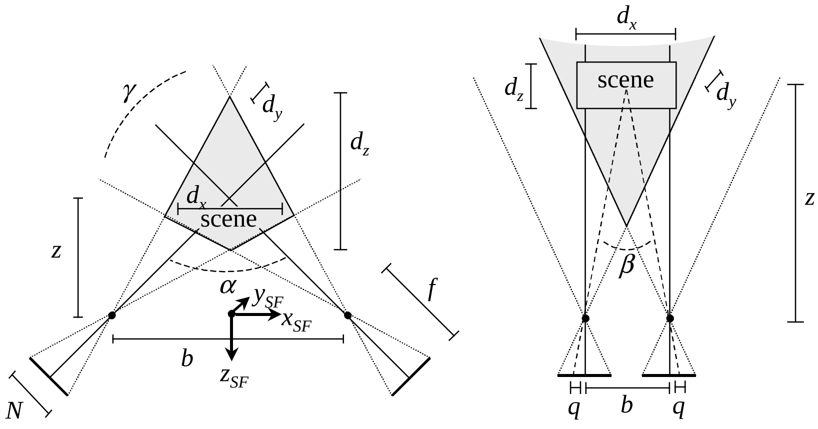
Our analysis provides a reasonably simple indication of the scene properties using our physical camera models. The position and size of the scene depend directly on the camera parameters. The resolution of the scene depends on the number of (more or less independent) corresponding pixel pairs in the stereo image. We assume that the number of correspondences is equal to the number of pixels in one image, i.e. we have a pixel-dense correspondence field (see Chapter 4). The accuracy is determined by the noise present in the estimated parameters and correspondences. The noise in the correspondences is found experimentally in Chapter 3 when detecting special markers in images (only for fixed calibration) and determined by the correspondence algorithms in Chapter 4. The amount of noise in the parameters is found experimentally in Chapter 3 on camera calibration.

In section 2.6.1 we first describe a simple camera model that we use to keep the analysis comprehensible. Then in section 2.6.2 we determine the position P, size d and resolution Δd of the scene. That will help us in sections 2.6.3 and 2.6.4 to deal with the 3-D acquisition accuracy Δa due to errors in the correspondences and the parameters respectively. In section 2.6.4 we will consider the additional aspects when using the more complex general camera model.

2.6.1 Simple stereo camera model

In order to avoid a very complex analysis, we will use a simple stereo camera model, shown in Figure 2.16. For the internal parameters, we assume that the lenses have no distortion, that the CCD chips have no skew and have a size of N x N pixels and are positioned without any error, that the focal lengths are both equal to f, and that the pixel aspect ratios are 1.

From the external parameters we consider only the baseline b and convergence angle α. The baseline b and camera-scene distance z are given in meters, while f is given in hpu.



(a) ***(b)***

Figure 2.16 A simple camera model for the analysis of the scene properties (top view or x_{SF} , z_{SF} plane), ***a)*** convergent setup, ***b)*** parallel setup.

Figure 2.16a shows the case of large convergence ($\alpha \approx 90°$) and Figure 2.16b shows the case of zero convergence ($\alpha = 0°$). The latter equals the parallel camera setup from section 2.4.9. Clearly, only a limited part of the viewing angles of the parallel cameras overlap to provide stereo viewing of the scene. This is the price we have to pay for the advantages of this setup. In Chapter 6 we will see that there are methods to circumvent this drawback.

For the parallel setup, the angle between triangulated light rays is equal to β, which differs for each scene point. For the convergent setup, the angle is approximated by α for all scene points. The viewing angle of the cameras is γ:

$$\tan \tfrac{1}{2}\gamma = \frac{N}{2f} \tag{2.50}$$

For small γ, we find:

$$\gamma \approx \frac{N}{f} \tag{2.51}$$

2.6.2 Position, size and resolution of the scene

We will examine the position P_{center}, size d_x, d_y, d_z and resolution Δd_x, Δd_y, Δd_z of the scene for the convergent and parallel camera setups separately.

Convergent cameras

Figure 2.16a shows that the scene is approximately located around the intersection of the optical axes. For this we find:

$$P_{center}^{\sigma_{SF}} = \begin{bmatrix} 0 \\ 0 \\ -z \end{bmatrix} \qquad z = \frac{b}{2\sin\frac{1}{2}\alpha} \tag{2.52}$$

The shape of the scene is similar to a kite. To simplify the description of its size, we define a bounding box around the scene as illustrated in Figure 2.17. For the size of this box we find after some calculations:

$$\begin{aligned} box_x &= b\left(1 - \frac{\tan\frac{1}{2}(\alpha-\gamma)}{\tan\frac{1}{2}\alpha}\right) \approx \tfrac{1}{2}b\gamma\frac{1}{\sin\frac{1}{2}\alpha\cos\frac{1}{2}\alpha} \\ box_y &= b\sqrt{1+\frac{1}{\tan^2\frac{1}{2}\alpha}}\tan\tfrac{1}{2}\gamma \approx \tfrac{1}{2}b\gamma\frac{1}{\sin\frac{1}{2}\alpha} \\ box_z &= \tfrac{1}{2}b\left(\frac{1}{\tan\frac{1}{2}(\alpha-\gamma)} - \frac{1}{\tan\frac{1}{2}(\alpha+\gamma)}\right) \approx \tfrac{1}{2}b\gamma\frac{\cos^2\frac{1}{2}\alpha}{\sin^2\frac{1}{2}\alpha\cos^2\frac{1}{2}\alpha - \frac{1}{4}\gamma^2} \end{aligned} \tag{2.53}$$

As can be seen in Figure 2.17, the volume of the bounding box is twice the volume of the scene, due to the fact that the scene is not a square aligned with the x_{SF} , z_{SF} axes. Therefore, we approximate the scene shape by a box with size:

$$\begin{aligned} d_x &= \tfrac{1}{2}\sqrt{2}box_x \\ d_y &= box_y \\ d_z &= \tfrac{1}{2}\sqrt{2}box_z \end{aligned} \tag{2.54}$$

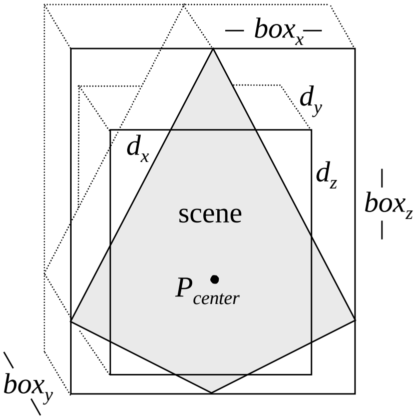


Figure 2.17 Scene size and bounding box for convergent camera setup.

For small γ, (2.53) and (2.54) become:

$$\begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \frac{b\gamma}{2\sin\frac{1}{2}\alpha} \begin{bmatrix} \dfrac{1}{\sqrt{2}\cos\frac{1}{2}\alpha} \\ 1 \\ \dfrac{1}{\sqrt{2}\sin\frac{1}{2}\alpha} \end{bmatrix} \tag{2.55}$$

Since the CCD chip has a viewing angle γ, each pixel has a viewing angle of more or less γ / N. If a corresponding pixel pair is triangulated, the intersection yields a cube that has a similar orientation as the entire scene but whose edge length is N times smaller. This is the resolution of the scene:

$$\Delta d = \frac{1}{N} d \tag{2.56}$$

This gives a total number of N^3 elementary 3-D scene volumes, generally called voxels. This seems much more than the $2N^2$ (left and right) pixel luminances we started with to acquire the 3-D scene. We have to keep in mind, however, that only scene surfaces can be captured by stereo cameras. If $2N^2$ pixels yield N^2 corresponding pairs, only N^2 out of the N^3 voxels get assigned some scene point (or slightly more due to interpolation), while the other voxels contain empty space (foreground) or remain undefined (behind scene surface).

For $\alpha \approx 90°$, we find that the scene has about the same size in all directions. It is a cube, oriented at 45° with the x and z axes. Its shape is approximated by a similar cube aligned with the x and z axes. Its size is:

$$d = \tfrac{1}{2}\sqrt{2} b\gamma \tag{2.57}$$

And by using (2.51) and (2.56) we obtain:

$$\Delta d = \tfrac{1}{2}\sqrt{2}\frac{b\gamma}{N} \approx \tfrac{1}{2}\sqrt{2}\frac{b}{f} \tag{2.58}$$

Concerning the last term of (2.58), we point out that if more pixels are added on the same CCD chip area, then N increases, so the hpu unit will become smaller and f will increase, producing the expected decrease in Δd.

Parallel cameras

In the parallel case, we see in Figure 2.16b that the position and size of the scene are not fully restricted. We assume that the scene is a box, centered at:

$$P_{center}^{\sigma_{SF}} = \begin{bmatrix} 0 \\ 0 \\ -z \end{bmatrix} \qquad z > \frac{bf}{N} \approx \frac{b}{\gamma} \tag{2.59}$$

The depth position z and the depth size d_z are free parameters. The scene size d_x and d_y are restricted by a function of z:

$$\begin{aligned} d_x(z) &< \frac{zN}{f} - b \approx z\gamma - b \\ d_y(z) &< \frac{zN}{f} \approx z\gamma \end{aligned} \tag{2.60}$$

For $z = bf/N$, the d_x becomes zero. At this depth, the viewing angles of the two cameras do not overlap at all. The relative overlap $0 \leq \eta \leq 1$ is a function of depth:

$$\eta = 1 - \frac{bf}{zN} = 1 - \frac{b}{z\gamma} \tag{2.61}$$

The choice of z, b, γ and their effect on overlap and thus the scene size plays an important role in the design of the PANORAMA multi-viewpoint system, see Chapter 6.

Similar to the converging camera setup, the resolution is determined by the fact that pixels have a ‘viewing angle’ γ/N, which is $1/N$ of the total viewing angle. Then, when we apply (2.56) to (2.60), this results in:

$$\Delta d_{x,y} = \frac{z}{f} \tag{2.62}$$

The overlap η has no influence on the size of the voxels. The Δd_z is obtained as follows, by using β, the convergence angle between light rays intersecting at depth z on the z_{SF} axis (see Figure 2.16b). The light rays originate from two points on the left and right CCD at a pixel distance q from the CCD center, see Figure 2.16b. For β we find:

$$\tan \tfrac{1}{2}\beta = \frac{b}{2z} = \frac{q}{f} \tag{2.63}$$

From the two right-hand sides we find:

$$z = \frac{bf}{2q} \qquad \left|\frac{\Delta z}{\Delta q}\right| = \left|-\frac{bf}{2q^2}\right| = \frac{2z^2}{bf} \tag{2.64}$$

For $\Delta q = 1/2$, we have a shift of half a pixel in both images, which is seen as one unit of resolution (similar to a shift of one pixel in only one image). This yields:

$$\Delta d_z = \frac{z^2}{bf} \tag{2.65}$$

And together with (2.62) we find:

$$\begin{bmatrix} \Delta d_x \\ \Delta d_y \\ \Delta d_z \end{bmatrix} = \frac{z}{f} \begin{bmatrix} 1 \\ 1 \\ z/b \end{bmatrix} \tag{2.66}$$

Thus, in the parallel setup the resolution drops down linearly with z in the x and y directions, and quadratic in the z direction. For $z \approx b$ the resolution is isotropic. Due to the overlap constraint $\eta > 0$, the scene can only be present at this position if a wide-angle (> 60°) camera is used with a focal length $f < N$.

2.6.3 Scene accuracy due to noise in correspondences

Here we will derive an indication of the accuracy Δa of the acquired scene given noise or errors in the estimated correspondences. The results will be related to the resolution Δd found in the previous section. First we will discuss the accuracy of the correspondences followed by the accuracy of the acquired scene.

Accuracy of correspondences

We assume that for each pixel in one of the images, a corresponding pixel in the other image can be found with an error in the order of 0.1 to 10 pixels. We assume that the correspondences are either uncorrelated with neighboring correspondences (neighboring pixels in the images) or correlated only in a small neighborhood. In such cases, the scene has its correct global shape but there is some local geometric noise in it.

Figure 2.18 illustrates the errors in the correspondences. In principle, the error can be modeled to reside fully in one of the pixels in the pair while the position of the other is by definition correct. To keep our analysis simple we will distribute the error symmetrically over both pixels, with zero mean and magnitude σ_{cor}. For both pixels we use the continuous I coordinates from section 2.3 to allow for sub-pixel accuracy correspondences.

The error can be isotropic in the x_I and y_I image coordinates, as is expected for general correspondence algorithms that have no preference for a specific direction. The error can also be constrained to one specific direction, in the case that correspondence estimation is performed after camera calibration. In that case, from the camera parameters we can derive the so-called epipolar geometry that determines these directions, see Appendix B.

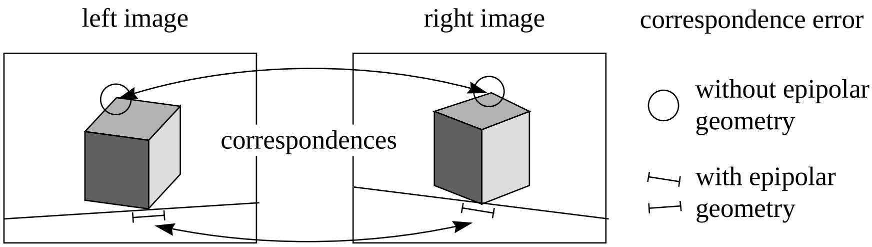


Figure 2.18 A corresponding pixel pair with estimation errors or noise.

Scene accuracy without epipolar geometry

If no epipolar geometry is used, the correspondence errors are isotropic. We may think of these errors as effectively scaling the viewing angles γ/N of the light rays originating from the corresponding pixel pair by σ_{cor} in both x_I and y_I directions. For the accuracy of the acquired scene we may then write:

$$\Delta a = \sigma_{cor} \Delta d \tag{2.67}$$

With epipolar geometry

As shown in Appendix B, in the parallel setup the scan lines are equal to the epipolar lines. That means that corresponding points share their y_I image coordinate, as was derived mathematically in section 2.4.9. Incorporating this constraint in the correspondence estimation algorithm yields $\sigma_{cor}^{y} = 0$. To provide a simple rule of thumb, we will only investigate the accuracy around the scene center, located on the z_{SF} axis. For this point, the epipolar plane coincides with the xz plane and we find:

$$\begin{bmatrix} \Delta a_x \\ \Delta a_y \\ \Delta a_z \end{bmatrix} = \sigma_{cor} \begin{bmatrix} \Delta d_x \\ 0 \\ \Delta d_z \end{bmatrix} \tag{2.68}$$

In the convergent case, the center scan line is also an epipolar line in both images (see Appendix B) and (2.68) also applies.

2.6.4 Scene accuracy due to noise in the parameters

We will examine the accuracy of the acquired scene as a function of small errors in the parameters, using the convergent camera model. First we examine the parameters from the simple model; baseline b, focal lengths f, their combined effects b and f, and the convergence angle α. For the parameters of the general model (lens distortion, non square pixels, more rotational angles and different focal lengths for the two cameras), we will describe the errors by equivalent errors in the correspondence field $\sigma_{cor\text{-}equi}$ and then use (2.67), the scene accuracy model for correspondence errors. The errors are dealt with separately without taking dependencies into account. Since their mutual effects may cancel out one another, the calculated errors correspond to worst-case errors. We assume skew, CCD misorientation and mispositioning to be zero.

Baseline

If the baseline b is estimated with an error Δb, then all acquired scene points are scaled in the SF frame by a factor $1+\Delta b/b$ around O_{SF}. Thus, this error has a global character with the following effects:

- Global translation of the acquired scene on a line towards or away from O_{SF}
- Global scaling
- Orientation and shape (all angles) are preserved

In self-calibration methods, $b = 1$ [bu]. If the acquired scene is interpreted in meters, a similar scale error appears.

Focal length

If the focal length f of the cameras is estimated with equal error Δf, the following can be seen from Figure 2.16 for the convergent setup:

- Position and orientation of the acquired scene remain fixed
- Global scale change by $1-\Delta f/f$
- Small deformations: when f is overestimated, the kite-like shape of the scene becomes more square

The Δf has an impact on γ via (2.51). Then (2.55) describes the scale change which is linear with γ. The deformations are described by the d_x and d_z terms of (2.55). For larger γ, (2.53) yields an additional anisotropic scaling in the z_{SF} direction, which is $1/(1-\frac{1}{2}\gamma^2)$ for $\alpha = 90°$.

Baseline and focal lengths

If the errors Δb and Δf are combined, they may cancel out one another’s scaling effects:

- Scale and orientation remain fixed
- Position may change
- Small deformations: the shape of the scene may change between kite-like and square

Convergence angle

The error $\Delta\alpha$ in the estimation of the convergence angle has the following effects:

- Position of the acquired scene changes in the z_{SF} direction
- Orientation remains fixed
- Small deformations: the shape of the scene may change between kite-like and square

The translation is described by (2.52) and the deformations by (2.55).

Lens distortion

If lens distortion is not modeled, the *K*s are zero and the errors in the *K*s are equal to the actual *K*s themselves. The resulting scene errors can be modeled similarly as in (2.67) due to errors in correspondence field. The σ_{cor} has to be replaced by an appropriate $\sigma_{lens\text{-}dist}$ that represents the lens distortion in pixels, e.g. 5-10 [Tsai87]. Unlike errors in the correspondence field, the lens distortion errors are distributed non-uniformly over the image (zero in the center) and highly correlated over the image. The scene will thus be correct in the center and deformed globally.

If lens distortion is modeled but the *K*s are estimated with error ΔK, then the appropriate $\sigma_{lens\text{-}dist}$ must be scaled with the average of the relative accuracies of the *K*s:

$$\sigma_{cor-equi} = \sigma_{lens-dist} \frac{1}{N_K} \sum_{\text{all } Ks} \left| \frac{\Delta K}{K} \right| \tag{2.69}$$

In our model, $N_K = 4$ (K_3 and K_5 for each camera).

Pixel ratios

For errors $\Delta s_{y;L}$ and $\Delta s_{y;R}$ in the pixel aspect ratios $s_{y;L}$ and $s_{y;R}$, the equivalent correspondence error is:

$$\sigma_{cor-equi} = \tfrac{1}{2} N \left(\frac{\Delta s_{y;L}}{s_{y;L}} + \frac{\Delta s_{y;R}}{s_{y;R}} \right) \tag{2.70}$$

This error holds at the top and bottom scan line of the CCD, where it is maximal. In the center scan line, the error is zero. For $N = 1000$, $s_y = 1$ and $\Delta s_y = 0.01$, the equivalent maximum error is 10 pixels.

Two focal lengths

For errors Δf_L and Δf_R in the focal lengths f_L and f_R, the equivalent correspondence error is:

$$\sigma_{cor-equi} = \tfrac{1}{2} N \left(\frac{\Delta f_L}{f_L} + \frac{\Delta f_R}{f_R} \right) \tag{2.71}$$

This error holds at the border of the CCD, where it is maximal. In the center, the error is zero. For $N = 1000$, $f = 1000$ and $\Delta f = 10$, the equivalent maximum error is 10 pixels.

General rotation angles

If the rotations around the optical axes of the cameras, $\varphi_{SF}^{LFL;z}$ and $\varphi_{SF}^{LFR;z}$, have errors of $\Delta\varphi_z$, the equivalent correspondence error is:

$$\sigma_{cor-equi} = N \Delta\varphi_z \tag{2.72}$$

This error is holds at the border of the CCD, where it is maximal. In the center of the CCD, the error is zero. For N = 1000, an error of $\Delta\varphi_z$ = 1° already produces maximum errors of about 17 pixels.

The viewing angles of both cameras around the baseline, $\varphi_{SF}^{LF;x}$, are coupled via (2.3). If one camera rotates upwards, the other rotates downwards. In a first approximation, this yields only intersection errors via (2.48) in the triangulation process, but no acquisition errors via (2.47). The equivalent correspondence error is then approximated by zero:

$$\sigma_{cor-equi} = 0\Delta\varphi_x \tag{2.73}$$

The two angles $\varphi_{SF}^{LF;y}$ determine the position of the acquired scene in the x_{SF} and z_{SF} directions. In the simple model, this was done by α, which could only move the scene along the z_{SF} axis. In the general model, if both angles contain errors, the global position of the scene will change both in x_{SF} and z_{SF} directions, but the orientation and shape will remain practically the same. This produces an equivalent systematic error in the correspondences, but the random error can be approximated by zero:

$$\sigma_{cor-equi} = 0\Delta\varphi_y \tag{2.74}$$

2.7 Conclusions

In this chapter we described a general model for stereo cameras. As a huge amount of literature on this topic already exists, the main goal was to unify these models in their notation, their model properties and the ability to estimate their parameters by camera calibration.

A new notation was introduced, borrowed from tensor notation in physics, which helps to keep the models comprehensible despite the large number of variables used. We showed the relations between our model and many other models from literature, aided by the new notation.

A single, general camera model was designed containing only parameters that correspond to physical measurable quantities such as lengths and angles. This makes it easier to model the cameras and to verify the estimated parameters. The same model can be used both for fixed and self-calibration methods.

Lens distortion has been dealt with in detail; we focused on both modeling and computational properties. On the one hand, lens distortion plays an important role in enhancing the quality of the model of any practical camera. On the other hand, we conjectured that the ‘artifact’ lens distortion might be used to our advantage in self-calibration approaches. A theoretical proof exists which says that cameras without lens distortion cannot be calibrated accurately unless the model has 7 or less parameters. With such a simple camera model, scenes cannot be acquired in a geometrically correct and accurate way. However, it is not known whether this proof also holds if lens distortion is present in the cameras as well as in the camera model. We will be able to test this in the next chapter with our model that includes lens distortion.

Less complex models were derived from the general model, and compared with similar models from literature. These include e.g. pinhole models, in which the lenses are assumed to be ideal, i.e. without distortion. We discussed the parallel camera setup in detail, to be used in the PANORAMA system in Chapter 6.

Finally, to acquire the scene, we must triangulate corresponding pixels from the left and right image from the stereo pair (to be estimated in Chapter 4). The triangulation process accompanying our model was described in detail, as well as the properties of the resulting scene. These include the position, the size, the accuracy and the resolution of the scene, for which simple rules of thumb were derived. The results can be used before scene recording to make proper choices for the camera setup.

The work done in this chapter is meant as a first step towards scene acquisition. The next chapter deals with stereo camera calibration, which makes use of the camera model defined in the current chapter. After correspondence estimation in Chapter 4, we will be able to apply the triangulation process we defined in the current chapter. For the PANORAMA system design in Chapter 6, we have discussed the parallel camera setup in detail and found simple guidelines to determine the scene properties.

Chapter 3

Stereo camera calibration

3.1 Introduction

This chapter deals with the calibration of the stereo camera. Figure 3.1 shows its role in the scene acquisition process.

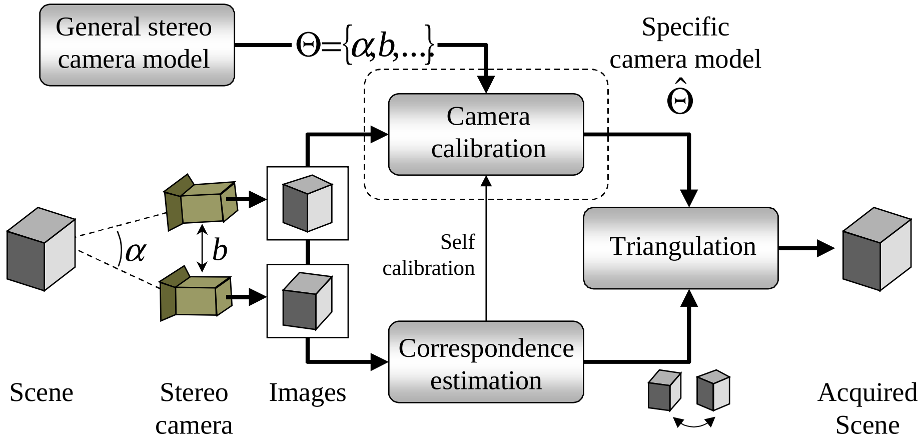


Figure 3.1 The scene acquisition process. This chapter deals with the camera calibration algorithm (shown dotted).

In Chapter 2, we designed a general, parameterized camera model. The task of camera calibration in this chapter is to estimate the values for these parameters. The parameters describe a single camera model that is specific for the actual cameras.

There are two major kinds of camera calibration. The first is fixed calibration, which originates from photogrammetry [Brow71, Slam80]. Shortly after, it was introduced in the computer vision literature, where it is often referred to by the more general term calibration [Wolt78, Tsai87, Weng92, Wei94] and also by strong calibration [Pede99]. In fixed calibration we make use of a highly controlled scene before recording and processing the actual scene. Figure 3.2 outlines the calibration procedure. The controlled scene consists of a rigid object with some special markers on it. The geometric and photometric properties of the markers are known precisely, so they can be located in both images. In this way a corresponding point pair is found for each of the markers. The general principle of fixed calibration is to search for those camera parameters for which triangulation of all corresponding marker-point pairs gives an accurate reconstruction of the calibration object. In that case, the camera model represents the actual cameras accurately.

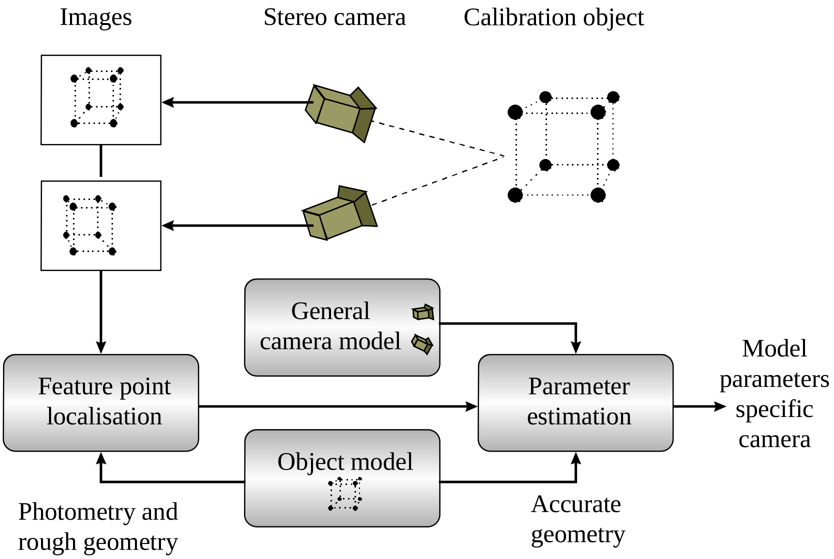


Figure 3.2 Fixed calibration.

Fixed calibration has two drawbacks. First, it needs user interaction, since a special object has to be manufactured and recorded. Secondly, the cameras may not undergo any change in between the recordings of calibration object and actual scene. Each time a change is made in the camera setup, such as zooming in or out, the calibration has to be repeated.

Many cases exist in which we cannot use fixed calibration: scene acquisition from ancient stereo photos, from very small or very large scenes for which no calibration object can be made, in hazardous environments where user interaction is unwanted or impossible, or for dynamical scenes that require the cameras to undergo changes during recording (zoom, orientation). In such cases, we must resort to a calibration method that uses only the images of the actual scene.

Currently, research is focusing on self-calibration [Arms96, Rede98d, Faug92], sometimes referred to as autocalibration [Ziss98], active calibration [Basu95] or blind calibration [Pede99]. In self-calibration, the calibration is performed using images of the actual scene. First corresponding pixels between the left and right image are estimated, see Figure 3.1 (described in Chapter 4). The fact that the light rays of corresponding pixels must intersect, even if we do not know exactly at which position in space, provides a constraint on the camera parameters enabling their estimation. This does not require a special object, relieves the user of any interaction [Poll98, Rede98d] and allows for continuous camera changes during recording. In [Pede97a, Pede99] a calibration technique is described that employs a calibration object with a geometry that is only roughly known, but is refined (re-calibrated) during the calibration of the cameras. It lies more or less in between the fixed and self-calibration methods.

The correspondences needed for self-calibration are estimated using some photometric/geometric model of the scene. These are e.g. the Constant Image Brightness constraint (scene point emits light uniformly in all directions, see Chapter 4) or the presence of corners [Poll98], or smooth scene surfaces [Rede98d]. These are very general models for a very large set of possible (natural) scenes, but still, it contradicts the statement that self-calibration does not require a priori information about the scene [Arms96, Luon93, Poll98]. In structure-from-motion applications, where scenes are acquired by processing of multiple images from a single camera, similar prior knowledge of the scene is required [Jeba99].

The scene models in self-calibration encompass all natural scenes. Such general models are much less informative or accurate compared to the object models used in fixed calibration, that contain highly detailed scene information. Therefore, it is much harder to obtain the parameters in self-calibration. First, since we have no reference to the standard meter, the absolute scale of the scene cannot be obtained. Secondly, it can be proven that if self-calibration is applied on cameras with ideal lenses, we can measure at most 7 parameters [Arms96, Csur97, Luon93]. Any stereo camera model that has more parameters results in incomplete parameter estimation. For successful calibration, we thus need extra constraints or knowledge [Deve96], such as a known pixel aspect ratio [Poll98].

In this chapter, our contributions to the field of camera calibration are the following. We will

- derive a marker detection scheme for fixed calibration that is fully automatic and accurate up to 0.1-0.01 pixel by incorporating curvature effects due to perspective and lens distortion.
- unify fixed and self-calibration methods.
- use the Bayesian probability framework throughout the calibration procedure.
- apply simulated annealing as a general search algorithm in the calibration, which spares the designer analytic work and provides high flexibility in choosing a camera model.
- show that we can measure more than 7 camera parameters by self-calibration, provided that lens distortion is present in the cameras and their model.

The majority of the chapter deals with fixed calibration in detail. From that point on, the step towards self-calibration can be made with minimal effort. Section 3.2 describes the calibration object for fixed calibration. We will outline the marker extraction procedure globally in section 3.3, while the details can be found in Appendix D. We validate the marker detection algorithm by experiments in section 3.4. The parameter estimation algorithm for fixed calibration is then explained in section 3.5, followed by experiments in section 3.6. We will derive the self-calibration algorithm in section 3.7, followed by experiments in section 3.8. Finally, section 3.9 concludes the chapter.

3.2 Calibration object

The calibration object should contain a number $N_{markers}$ of marker points P_i that have a very well-known geometric model and photometric model. A theoretical example of a perfect calibration object is a transparent box made out of glass, with a large number of small spheres, uniformly distributed in the box volume with known positions and different colors. The centers of the spheres are the marker points P_i. Then, if the spheres do not occlude one another, they can easily be detected in the images by color segmentation. The apparent 2-D center is then an approximation of the real projection of the 3-D sphere center.

The geometric model of the calibration object must be given in some reference frame; we call it the calibration reference frame *CF*. The position and orientation of this frame can be chosen arbitrarily with respect to the object. The geometry of the points is then given by:

$$P_i^{\sigma_{CF}} \tag{3.1}$$

In our scheme, the *CF* frame is not related to the camera system and thus has to be estimated along with the camera parameters, introducing 6 additional parameters. In many other schemes, the scene frame *SC* is chosen to be *CF* rather than *SF*. This avoids the introduction of the 6 extra variables, but the camera model itself then contains 12 external parameters instead of 6, as in our case (see section 2.3.2).

In order to obtain parameters that yield an accurate acquisition of the actual scene, we must ensure that the calibration object occupies all 3-D space that will be occupied by the scene later on [Pede99]. Accurate manufacturing of such possibly large objects is very expensive. Often a much simpler calibration object is used, such as a planar object. The 3-D scene space is then filled by recording the plane in a number N_{view} of different positions and orientations. This yields a compound calibration object with $N_{markers}=N_{view}N_{plate\text{-}markers}$ marker points P_{ij} in 3-D space. In [Tsai87] it is found that no specific orientations of the plane are required for the compound object (a number of coplanar orientations differing only in position suffices). Each of the views also yields a different CF_j frame, and the coordinates of the points are now only known through:

$$P_{ij}^{\sigma_{CF_j}} = \delta_{\sigma_{CF}}^{\sigma_{CF_j}} P_i^{\sigma_{CF}} = P_i^{\sigma_{CF}} \tag{3.2}$$

The δ represents the fact that changing the orientation of the plate in 3-D space does not change its model in the *CF* frame. In the last equality in (3.2) it is defined that the σ_{SF} and $\sigma_{SF;j}$ indices run over $\{x,y,z\}$ simultaneously.

In the end, we need to know the 3-D positions of all points in the scene frame (*SF* in our case). Therefore, we must estimate the position *O* and orientation *V* of all CF_j frames along with the camera parameters. This introduces $6N_{view}$ parameters in addition to the $N_{cam\text{-}model}$ parameters from the stereo camera model. We then have:

$$P_{ij}^{\sigma_{SF}} = V_{\sigma_{CF_j}}^{\sigma_{SF}} P_i^{\sigma_{CF}} + O_{CF_j}^{\sigma_{SF}} \tag{3.3}$$

Our calibration object is a planar, black object with $N_{plate\text{-}markers} = 48 = 8\text{x}6$ white circular markers placed on a simple grid, see Figure 3.3. The feature points P_i are the centers of the markers.

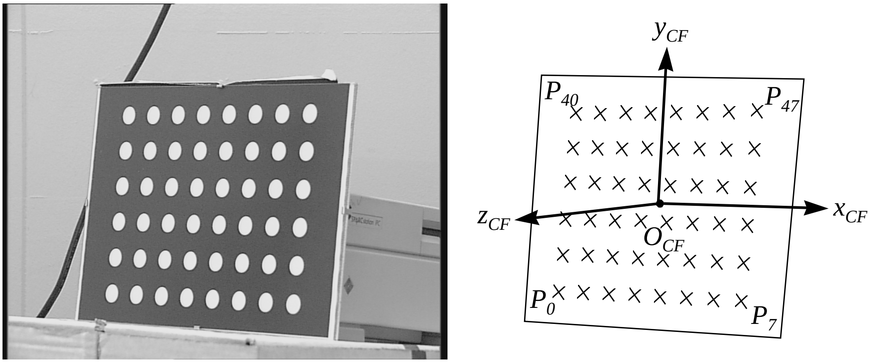


Figure 3.3 The calibration object.

Two versions of the object are available, see Table 3.1. Their *CF* frames are defined such that the origin is in the center of all markers. The orientation is chosen such that the z_{CF} axis is orthogonal to the plane and the x_{CF} and y_{CF} are aligned with the markers. The marker index starts at the bottom left and runs from left to right, from bottom to top, ending in the top right corner. The object is symmetric under 180° rotation.

Global size	center-to-center	marker diameter	type	Accuracy
A1	10 cm	4 cm	professional	*xy* 10 µm, *z* 20 µm
A4	3 cm	1.5 cm	laser printer	*xy* 0.2 mm, *z* better

Table 3.1 Two versions of the calibration object.

The A1 plate is a professional plate which has been calibrated by the manufacturer. We made the A4 plate ourselves by printing the marker pattern with a Hewlett Packard 5SiMX laser printer. The sheet of paper was put in cheap planar glass frame. We calibrated the plate manually with an accurate Mitutoyo ruler.

3.3 Marker detection

Our marker detection scheme is fully automatic, extremely robust and very accurate, outperforming all current schemes in literature. The procedure is done separately for the multiple views of the calibration object as well as for the left and right images. It involves seven steps:

- Finding regions that possibly contain a marker.
- Design of a parameterized ellipse model of a single marker inside a region.

- Estimation of the photometric parameters for each region (e.g. *SNR*).
- Detecting valid regions by a check on the photometric parameters.
- Estimation of the ellipse position within each region.
- Relating each ellipse with a marker on the plate by sorting all ellipses globally in the 8x6 grid, additionally discarding the remaining falsely detected regions.
- Estimation of the marker centers P_i by incorporating curvature effects due to lens and perspective distortion.

Figure 3.4 illustrates the steps. The justification for so many steps is that we are faced with a task that seems quite simple, but in fact incorporates all aspects of camera calibration all at once. Since the camera parameters are unknown at this moment, the markers may appear in the images as circles, ellipses (due to slanted views) or even other shapes (due to lens distortion). From their photometric appearance we only know that there is a substantial contrast between the markers and the plate. The images may contain noise and may be slightly defocused (edges of markers less sharp). The scheme must be robust against or invariant to all these effects. Therefore, in each step we either make mild assumptions about the effects or try to estimate them along with the markers, without estimating actual camera parameters as this should be done by the calibration scheme.

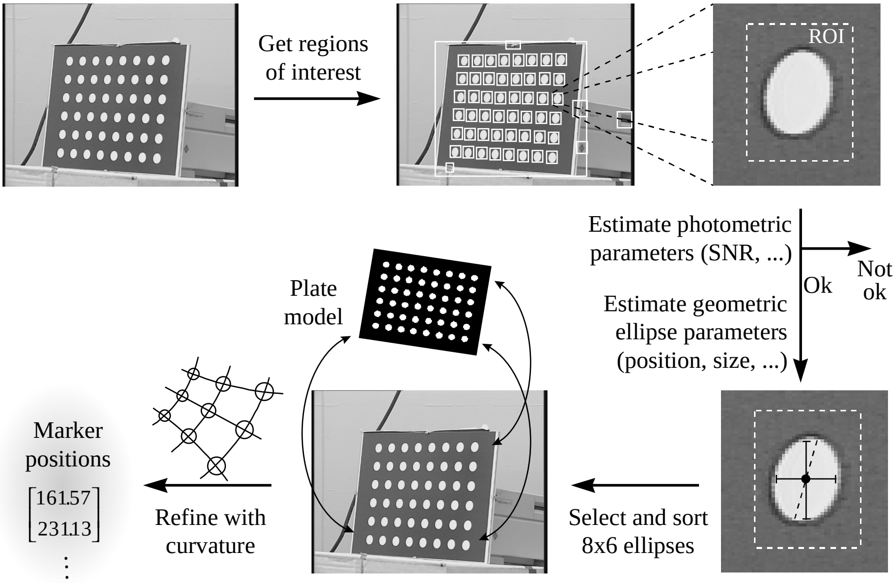

Get regions of interest
ROI
Estimate photometric parameters (SNR, ...)
Not ok
Ok
Estimate geometric ellipse parameters (position, size, ...)
Plate model
Marker positions
161.57
231.13
Refine with curvature
Select and sort 8x6 ellipses

Figure 3.4 The marker detection scheme.

The algorithm contains two new localization refinement methods. After the markers are roughly found by the center O_R of each region R, a well-known method for finding the marker center is the ‘center of gravity’ method, which determines the average of the image coordinates in the region weighted with luminance, here denoted by W_{lum}. Our first improvement uses the fact that the marker shape is (almost) an ellipse: we replace the luminance weight by a refined weight $W_{ellipse}$ that is less sensitive to noise. After sorting the markers, we perform our second new refinement by determining the marker centers while incorporating curvature effects due to lens and perspective distortions. In [Heik97] it was first shown that taking into account curvature improves the results. In the next section we will see that our algorithm outperforms all algorithms currently available.

A detailed description of the algorithm can be found in Appendix D.

3.4 Marker detection experiments

We evaluate the accuracy of the marker detection scheme by producing synthetic images of a synthetic calibration object. This gives us the ground truth values of the marker positions in 2-D image coordinates.

The synthetic plate has the properties of the A4 plate from Table 3.1. The adopted camera model is the general model from Chapter 2 in Table 2.3, of which only one camera will be used. For several sets of camera parameters and plate positions and orientations, we have constructed synthetic images by ray tracing. This procedure is described in section 2.5 (about triangulation), except that here the light rays are not triangulated with a ray from a second camera, but intersected with the synthetic plate. In the ray-tracing procedure, five additional elements are taken into account:

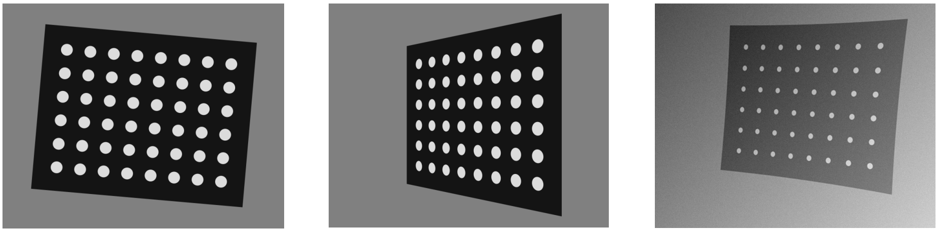
- For each image pixel, N_{over}xN_{over} rays are traced distributed over the pixel area, and their intensities are averaged. This models the different light rays that reach a single pixel on the CCD chip of a real camera.
- The image is linearly filtered by a uniform smoothing filter with size N_{smooth}xN_{smooth}, and a 3x3 edge enhance filter $[-\delta_{edge} \quad 1+2\delta_{edge} \; -\delta_{edge}]\,[-\delta_{edge} \; 1+2\delta_{edge} \; -\delta_{edge}]^T$. This models slight defocus of the lens and deliberate image enhancement filters in actual cameras.
- The brightness of the background and plate are not uniform as we added $\Delta I = bx_I + cy_I$. This models specular reflectivity of light on the plate.
- Gaussian noise is added with σ_{noise}. This models thermal noise in the CCD and further signal processing in the camera.
- The luminance is discretized to 0-255, modeling the discretization in cameras that provide digital images.

The images have CCIR601 size with $N_x = 720$ and $N_y = 576$. Figure 3.5 shows some images generated. Table 3.2 shows the results of the detection algorithm. The experiments were performed with the same settings for the marker detection algorithm. The systematic and random errors are shown, given by

$$\varepsilon_{sys} = |\mu_x| + |\mu_y| \qquad \varepsilon_{rnd} = \sqrt{\sigma_x^2 + \sigma_y^2} \tag{3.4}$$

where μ_x , μ_y is the marker error averaged over the 48 markers, and σ_x , σ_y the standard deviation. The results are given for four stages in the algorithm; the region center O_R, the ellipse center measured via W_{lum} and via $W_{ellipse}$, and the final result of the marker center with curvature compensation (see Appendix D).

In experiment A we used a perfect pinhole camera model, the calibration plate was exactly parallel to the CCD, N_{over} = 16, N_{smooth} = 1, δ_{edge} = 0, a = 20 (plate luminance), k = 200 (marker contrast), b = c = 0 and σ_{noise} = 0. For each of the other experiments, the differences with a previous experiment are described. Figure 3.5 shows the images used for experiments A, B and N.



(a) **(b)** **(c)**

Figure 3.5 Synthetic images to evaluate the marker detection algorithm, ***a)*** experiment A, ***b)*** experiment B, ***c)*** experiment N.

From all experiments, we observe in general that:

- The region detection works robust and with a constant accuracy of about 1-2 pixels.
- The conventional W_{lum} approach provides about 0.1 pixel accuracy.
- The $W_{ellipse}$ refinement method works about 2 times more accurately than with W_{lum}.
- The curvature algorithm improves the results 10 to 100 times, mostly due to a reduction of the systematic error.

Further, from experiments C-D we see that discarding the oversampling of the synthetic images during rendering limits the accuracy of the measured marker positions. The effect is only noticeable when curvature is taken into account. From experiments B-E and M-N we observe that making the markers smaller may enhance accuracy. This can be explained by the fact that curvature effects become quadratically less in size at smaller scales. However, if the markers' size is reduced more and more, the number of pixels in the regions will get smaller and image noise will prohibit any further gain in accuracy. From experiments C-H we observe that image smoothing has a positive effect on the determination of the ellipse center, but a negative effect on curvature refinement. This can be explained by the fact that the W_{lum} and $W_{ellipse}$ refinement procedures estimate the luminance center of the marker, of which the exact position is not affected by smoothing. The curvature refinement suffers from smoothing, since it assigns a different weight (G) to each pixel. The smoothing then exchanges luminance among pixels with different weights.

	Description	O_R		W_{lum}		$W_{ellipse}$		Curvature	
		ε_{sys}	ε_{rnd}	ε_{sys}	ε_{rnd}	ε_{sys}	ε_{rnd}	ε_{sys}	ε_{rnd}
A	Frontal view	1.75	0.45	0.000	0.057	0.000	0.000	0.000	0.001
B	Large plate slant A with $\varphi_y = 45°$	1.38	0.45	0.13	0.084	0.12	0.038	0.001	0.007
C	Normal plate slant A with $\varphi_x = \varphi_y = 10°$	1.41	0.46	0.080	0.056	0.076	0.005	0.000	0.001
D	No sampling within pixels C with $N_{over} = 1$	1.41	0.43	0.079	0.056	0.079	0.006	0.010	0.055
E	B with 50% smaller markers	1.46	0.49	0.021	0.096	0.030	0.009	0.000	0.002
F	Low contrast high noise, 14 dB C & $a = 70$, $k = 100$, $\sigma_{noise} = 10$	1.41	0.45	0.088	0.059	0.072	0.029	0.011	0.028
G	Nonuniform background C with $b = c = 1$	1.41	0.46	0.080	0.055	0.075	0.005	0.000	0.002
H	Image smoothing C with $N_{smooth} = 3$	1.37	0.47	0.007	0.056	0.076	0.028	0.024	0.022
I	Image edge enhancement C with $a = 70$, $k = 100$, $\delta_{edge} = ¼$	1.37	0.45	0.076	0.052	0.075	0.020	0.003	0.019
J	Effect of pixel ratio and skew B with $s_y = 0.8$ and $\theta = 5°$	1.36	0.46	0.13	0.072	0.12	0.039	0.002	0.007
K	Effect of lens distortion B with $K_3 = 1$ and $K_5 = ½$	1.31	0.51	0.15	0.16	0.15	0.16	0.001	0.030
L	K with 50% smaller markers	1.45	0.47	0.049	0.086	0.037	0.039	0.001	0.008
M	All effects, B with $K_3 = K_5 = ½$, $s_y = 0.9$, $\theta = 1°$ CCD mispositioning [20,30], misorientation [5°,4°], $a = 70$, $k = 100$, $\sigma_{noise} = 5$	1.46	0.44	0.076	0.084	0.077	0.050	0.063	0.022
N	M with 50% smaller markers	1.45	0.41	0.025	0.084	0.025	0.029	0.008	0.026

Table 3.2 Results of the marker detection algorithm. Systematic and random errors are shown in pixels, at four different stages in the algorithm: the region center O_R, the ellipse center measured with conventional W_{lum} and new $W_{ellipse}$ methods, and the final result of the marker center with curvature compensation.

For the final accuracy we find:

- The systematic errors are almost zero, compared to the random error.
- The errors are isotropic, i.e. the same in all directions.
- If all experiments are combined, an average error is found of about 0.01 pixel.
- In worst-case situations such as experiment M, the accuracy is still well below 0.1 pixel.

Figure 3.6 shows a scatter plot of the errors from all experiments. The mean is well below 10^{-3} pixels from the center, and the resultant $\sigma_{mrk} \approx 0.013$.

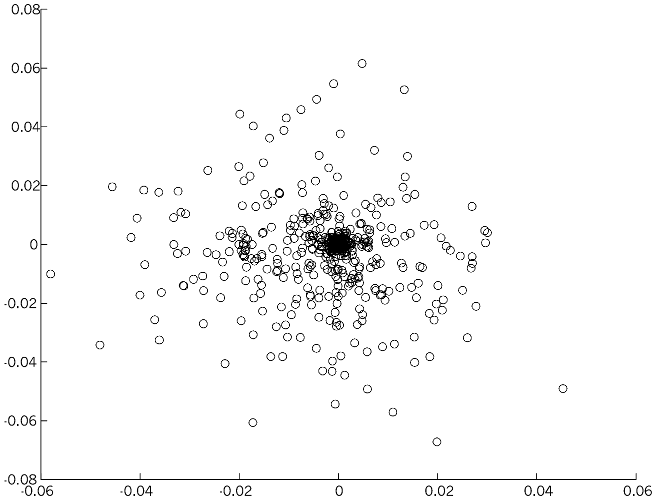


Figure 3.6 Scatter plot of the final marker errors from all experiments. It shows that the errors are isotropic and zero mean. The σ_{mrk} estimated from this plot is 0.013.

From subjective inspection of the marker position errors and more unsystematic experimental evidence, we believe that our results can be improved further. The automatic adaptation of the $A_?$ size with respect to the observed image noise and smoothing may be a future research area. Further, we observed that most of the errors in the low-noise experiments were concentrated in the outer ring of markers, in which no 3x3 neighborhood can be found during the curvature refinement (see Appendix D). Discarding these markers from a 10x8 grid may improve the results.

Other results from literature provide σ_{mrk} of 0.1 pixel in [Pede97a], 0.1-0.2 pixel in [Eela99b], 0.02 pixel in [Heik97], where also perspective distortion was included, 0.2 pixel in [Pede99] and 0.3-0.5 pixels in [Tsai87]. Clearly, our results outperform all of them, even in the extreme situation as depicted in Figure 3.5c, where small markers are visible in low contrast images with high noise, considerable lens distortion and specular reflections of the calibration plate.

3.5 Fixed calibration

In fixed calibration, the parameter estimation algorithm must find an estimate for all $N_{cam\text{-}model}$ parameters from the cameras as well as the $6N_{view}$ calibration plate positions and orientations, yielding N_{params} in total. Figure 3.7 shows the principle of solving this task. The input of the algorithm consists of the geometrical model of the calibration plate and the measured 2-D marker positions in the left and right images. The method searches for parameters that minimize the difference between the reconstructed and the actual calibration object in 3-D space. Using a first guess of all parameters, the algorithm positions the calibration object in *SF* space via (3.3), and reconstructs a calibration object from the measured markers by triangulation (Chapter 2). The difference between these two objects is calculated, e.g. the sum of all lengths of vectors going from a point on the actual object to the corresponding point on the reconstructed object. Then some search algorithm adjusts the parameters until the difference is minimized. Such algorithms can be analytical, giving the final answer in one step, or iterative, giving the answer or an estimate in a number of steps.

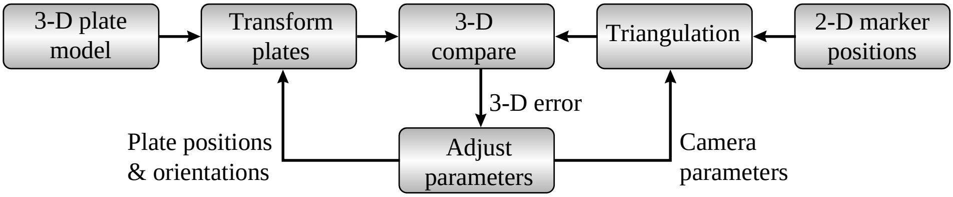


Figure 3.7 Parameter estimation by minimizing 3-D scene differences.

Figure 3.8 shows a method that is similar to the one above apart from the difference measure. The actual object is not only positioned in *SF* space, but it is also projected to the left and right images (see section 2.3). The difference is then calculated between measured 2-D marker positions and reconstructed 2-D marker positions.

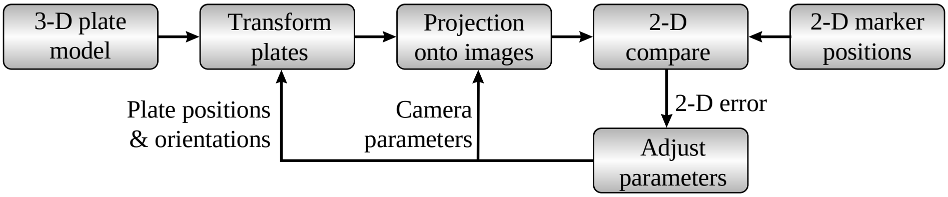


Figure 3.8 Parameter estimation by minimizing 2-D marker position errors.

In the next subsections, we will discuss the choice between the 2-D and 3-D methods, the formulation of the difference measure using the Bayesian probability framework, search algorithms for the minimization and finally how to evaluate or extract the accuracy of the estimated parameters.

3.5.1 Difference measure in 2-D or 3-D

We select the 2-D method for the following two reasons. First, we have a well-defined quantitative model for the differences in 2-D. Assuming that the camera model is appropriate, the 2-D differences are due to marker detection errors. For these differences, we have a good probabilistic model; they are modeled independently by a Gaussian, with $\mu = 0$ and $\sigma_{mrk} \approx 0.01$, found in section 3.4. In 3-D space, we can determine the difference between the actual calibration plate model and the plate reconstructed from the measured markers, but we do not have a quantitative probabilistic model for these differences. In [Roth97], a similar argument is made in favor of the 2-D method.

Secondly, the 2-D method can easily be extended to self-calibration schemes. The σ_{mrk} from the marker error model is then replaced by σ_{cor} from a correspondence estimation error model. A 3-D method cannot be applied in self-calibration, since then we have no object model and cannot determine the difference between the true and reconstructed object.

3.5.2 Formulation in the Bayesian probability framework

The Bayesian probability framework provides an elegant and general formulation tool. It allows us to formulate the 2-D difference measure and to integrate prior knowledge about the camera parameters. Only recently, the Bayesian approach has received attention in camera calibration [Pede99, Rede98d, Rede99b]. In our approach, we model the following parameters as random variables:

- Φ : all N_{params} parameters for the stereo camera and the positions and orientations of the calibration plates. The values of Φ are denoted by ϕ, the single parameters by ϕ_i.
- M : the $N_{markers}$ measured marker positions in the images. The marker detection algorithm supplies the values m of M. A single marker position is denoted by $\{ m_i^{x_I}, m_i^{y_I} \}$.

All models that contribute to the relation between Φ and M can be incorporated in a single joint probability density function (pdf) $p_{\Phi,M}$ (all parameters are continuous). We then define the ideal solution ϕ_{MAP} by the Maximum A Posteriori (MAP) criterion:

$$\phi_{MAP} = \arg\max_{\phi} p_{\Phi|M}\left(\phi, m\right) \tag{3.5}$$

The conditional pdf can be obtained from the joint pdf, but that is not necessary since m is a constant in the maximization (3.5):

$$\phi_{MAP} = \arg\max_{\phi} \frac{p_{\Phi,M}\left(\phi, m\right)}{p_M\left(m\right)} = \arg\max_{\phi} p_{\Phi,M}\left(\phi, m\right) \tag{3.6}$$

The joint model is designed by decomposing it in two parts:

$$p_{\Phi,M}\left(\phi, m\right) = p_{M|\Phi}\left(m, \phi\right) p_{\Phi}\left(\phi\right) \tag{3.7}$$

The p_{Φ} contains all our prior knowledge about the camera parameters, e.g. the prior camera model. The $p_{M|\Phi}$ contains the model that predicts where the marker detection algorithm will find the markers.

3.5.3 Prior model for camera parameters and plate positions

If we set p_Φ to a constant, we get a uniform prior probability for all possible camera and calibration object setups. With this, we avoid any bias towards a particular setup, which makes the procedure very generally applicable. However, some prior knowledge that is generally applicable is the fact that the baseline and focal lengths must be positive. We also know that the plates are positioned such that they are visible. Further, we always have some upper bound for parameters that model unwanted effects, such as lens distortion and CCD mispositioning. Incorporating this knowledge avoids spurious solutions.

We will introduce the following model for p_Φ:

$$p_\Phi(\phi) = \prod_{\substack{\text{all parameters} \\ i}} TG(min_i, max_i, \mu_i, \sigma_i, \phi_i) \tag{3.8}$$

The parameters are modeled independently by a truncated Gaussian (*TG*) probability density function (see Figure 3.9):

$$TG(min, max, \mu, \sigma, x) = \begin{cases} ke^{-\frac{(x-\mu)^2}{2\sigma^2}} & min < x < max \\ 0 & \text{else} \end{cases} \tag{3.9}$$

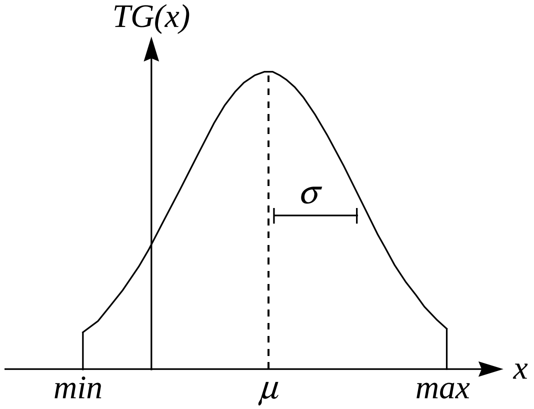


Figure 3.9 The truncated Gaussian *TG*.

The *TG* contains a minimum *min*, maximum *max*, mean μ and standard deviation σ. The *k* is a normalization constant. The *TG* enables the features shown in Table 3.3.

Table 3.4 shows the prior model for the general stereo camera model of Table 2.3 and all plate positions and orientations. Less complex models from section 2.4 can easily be implemented by setting the appropriate σ to zero.

σ	*min*	*max*	Parameter
∞	$-\infty$	∞	no prior model and completely unbiased
0	-	-	fixed to μ, is effectively eliminated from the model
σ	$-\infty$	∞	Gaussian with μ and σ
∞	*min*	*max*	Uniform between *min* and *max*

Table 3.3 The truncated Gaussian *TG*.

Parameters	*min*	*max*	μ	σ	explanation
b	0	∞	1	∞	baseline is always positive
$\varphi_{SF}^{LFL;x}$	-90°	90°	0	∞	viewing zones of camera must intersect
$\varphi_{SF}^{LFL;y}$ $\varphi_{SF}^{LFR;y}$	-90°	90°	0	∞	viewing zones of camera must intersect
$\varphi_{SF}^{LFL;z}$ $\varphi_{SF}^{LFR;z}$	-180°	180°	0	∞	camera rotation around the lens optical axis is free
$O_{PFL}^{z_{LFL}}$ $O_{PFR}^{z_{LFR}}$	0	∞	1000	∞	focal lengths are always positive
$\varphi_{LFL}^{PFL;xy}$ $\varphi_{LFR}^{PFR;xy}$	$-\infty$	∞	0	1	CCD mispositioning is small
$O_{PFL}^{xy_{LFL}}$ $O_{PFR}^{xy_{LFR}}$	$-\infty$	∞	0	1	CCD misorientation is small
$K_{3;L}$ $K_{5;L}$ $K_{3;R}$ $K_{5;R}$	$-\infty$	∞	0	0.1	Lens distortion is small
$s_{y;L}$ $s_{y;R}$	0	∞	1	0.1	Pixel aspect ratio is about 1
θ_L θ_R	$-\infty$	∞	0	0.1	CCD skew is small
$O_{CF_j}^{\sigma_{SF}}$	$\begin{bmatrix}-\infty\\-\infty\\-\infty\end{bmatrix}$	$\begin{bmatrix}\infty\\\infty\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\-1\end{bmatrix}$	$\begin{bmatrix}1\\1\\1\end{bmatrix}$	Plate position in view j, must be on the scene side $z_{SF} < 0$
$\varphi_{SF}^{CF_j;\sigma}$	$\begin{bmatrix}-90^\circ\\-90^\circ\\-180^\circ\end{bmatrix}$	$\begin{bmatrix}90^\circ\\90^\circ\\180^\circ\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\end{bmatrix}$	$\begin{bmatrix}\infty\\\infty\\\infty\end{bmatrix}$	Plate orientation in view j, mean is frontal view, must be visible gives *min*/*max*/σ

Table 3.4 Prior model for the general stereo camera model and plate positions. Discarding of a parameter is easily done by setting its $\sigma = 0$.

3.5.4 Predicting where the markers will be found

The $p_{M|\Phi}$ contains the model that predicts where the marker detection algorithm will find the markers. It consists of two parts. First, since Φ appears as prior knowledge in the p

subscript, we accept ϕ as the true camera and plate position parameters. Then we project the markers on the images according to section 2.3. The projections are the predicted marker positions m_{pred}. These are deterministically determined given ϕ. In the second step, we model that the marker algorithm finds the markers on the predicted positions detoriated by Gaussian noise with $\mu = 0$ and σ_{mrk}. We then obtain:

$$p_{M|\Phi}(m,\phi) = \prod_{\substack{i \\ \text{all markers} \\ \text{in all views,} \\ \text{left \& right} \\ \text{camera}}} \frac{1}{2\pi\sigma_{mrk}^2} e^{-\frac{\left(m_i^x - m_{pred;i}^x(\phi)\right)^2 + \left(m_i^y - m_{pred;i}^y(\phi)\right)^2}{2\sigma_{mrk}^2}} \tag{3.10}$$

3.5.5 Search algorithms for parameter estimation

The task of the search algorithm is to calculate (3.6), that is, the maximization of (3.7), consisting of (3.8) and (3.10). First, a practical challenge is the order of the probabilities. Since the dimensionality of the parameter space is in the order of 30, the values of the pdfs in such a large space are extremely small. If we have ϕ_{MAP} available except for one camera angle that is off-set by just 1°, the predicted markers m_{pred} in the image may shift by e.g. 5 pixels, which is about $500\sigma_{mrk}$. For two views of the calibration plate, 96 markers are off-set by $500\sigma_{mrk}$. The $p_{M|\Phi}$ will then be in the order of 10^{-10^6}. Such small numbers can be avoided by using energy $U = -\ln p$, which gives with (3.5)-(3.7):

$$\phi_{MAP} = \arg\min_{\phi} U_{\Phi,M}(\phi,m) = \arg\min_{\phi} U_{M|\Phi}(\phi,m) + U_{\Phi}(\phi) \tag{3.11}$$

For $U_{M|\Phi}$ we find

$$U_{M|\Phi}(m,\phi) = \sum_{\substack{\text{i} \\ \text{all markers} \\ \text{in all views}}} \left(\ln 2\pi\sigma_{mrk}^2 + \frac{\left(m_i^{x_I} - m_{pred;i}^{x_I}(\phi)\right)^2 + \left(m_i^{y_I} - m_{pred;i}^{y_I}(\phi)\right)^2}{2\sigma_{mrk}^2} \right) \tag{3.12}$$

And for U_{Φ}:

$$U_{\Phi}(\phi) = \sum_{\substack{\text{all parameters} \\ i}} \begin{cases} -\ln k + \dfrac{(\phi_i - \mu_i)^2}{2\sigma_i^2} & min_i < \phi_i < max_i \\ \infty & \text{else} \end{cases} \tag{3.13}$$

In our implementation, we will use some arbitrary large number N_{inf} to represent ∞. If we combine (3.11), (3.12) and (3.13) and discard constants, we obtain:

$$\begin{aligned}\phi_{MAP} = \arg\min_{\phi} \frac{1}{\sigma_{mrk}^2} & \sum_{\substack{\text{i}\\ \text{all markers}\\ \text{in all views}}} \left(m_i^{x_I} - m_{pred;i}^{x_I}(\phi)\right)^2 + \left(m_i^{y_I} - m_{pred;i}^{y_I}(\phi)\right)^2 \\ & + \sum_{\substack{\text{all parameters}\\ i}} \begin{cases} \dfrac{(\phi_i - \mu_i)^2}{\sigma_i^2} & min_i < \phi_i < max_i \\ N_{\inf} & \text{else} \end{cases}\end{aligned} \tag{3.14}$$

Many approaches are available for minimizations. Analytic formulations have been derived for camera calibration, e.g. [Boug98, Brow71, Faug92, Roth97, Truc98]. Their advantage is that the exact solution is found very fast, but they have many drawbacks. The analysis is generally quite complex and only in rare cases, non-linearities due to lens distortion are included [Tsai87]. Further, the method is not flexible in the sense that changing the camera model (e.g. discarding a parameter) may take quite some work. Also, they assume that there is only one single solution. If there are many (e.g. in case of a degenerate situation such as an orthographic camera), the approach may break down due to singularities.

Iterative approaches are also widely available. Their task is to wander around in the ϕ parameter space, interpreting (3.14) as just some function to minimize. In general, the exact solution is never reached, but a very good approximation can be obtained. Since little or no knowledge of the application domain (camera calibration) has to be put in the search algorithm, these methods save the designer much analytical work, enabling the introduction of e.g. lens distortion. If multiple best solutions exist, no singularities are encountered but an arbitrary choice is made among the best solutions.

The most well-known algorithm is gradient descent [Pres92], which walks along the gradient of U, hopefully towards the global minimum. In camera calibration, the Levenberg-Marquard (LM) method is popular [Arms96, Heik97, Stei97a], as it incorporates additional features to accelerate convergence [Pres92]. A disadvantage of these methods is that they require derivatives of U, which means considerable analytical work for the designer. If the camera model is changed, the work has to be done again. Further, these methods require a good inital guess of ϕ_{MAP} to avoid local minima and provide convergence to ϕ_{MAP} . The initial guess is mostly found by first applying an analytical method using a simplified camera model, e.g. without lens distortion [Heik97, Weng92, Zhan93]. In the area of Structure from Motion (SfM), where N cameras are calibrated one after another, (Extended) Kalman filters have been applied [Jeba99].

3.5.6 Our simulated annealing (SA) approach

We will use the stochastic search algorithm simulated annealing (SA) [Gema84, Pres92, Rede98d] to find the camera parameters ϕ_{MAP}. Its drawback is that it is computationally expensive, but it has several advantages. First, the method does not use any derivatives; only function evaluations of (3.14). Hence, no analytical work has to be done. This makes the approach very flexible, since the camera model can be changed very easily. In addition, SA algorithms start by looking in the parameter space globally, therefore we expect that it needs no initial guess.

The basic idea of SA is the following (see Figure 3.10). Based on a current estimate of the solution ϕ, a different candidate solution ϕ_{new} is obtained by adding a random (vector) perturbation to ϕ. Then, if $U_{M,\Phi}$ is lower for the new solution ($\Delta U < 0$), the new solution is accepted and will replace the current one. If U is higher ($\Delta U > 0$), the new solution is accepted with some low probability. The fact that now and then worse solutions are accepted enables the SA algorithm to escape from local minima. The acceptance probability is regulated by a so-called temperature T. The idea is to start the algorithm with a high initial temperature T_0. Then almost all perturbations are accepted, which allows the algorithm to wander in the solution space globally, without being dependend on a good initial solution ϕ_0. Then via some cooling schedule, the temperature is slowly lowered and the algorithm converges only to better solutions.

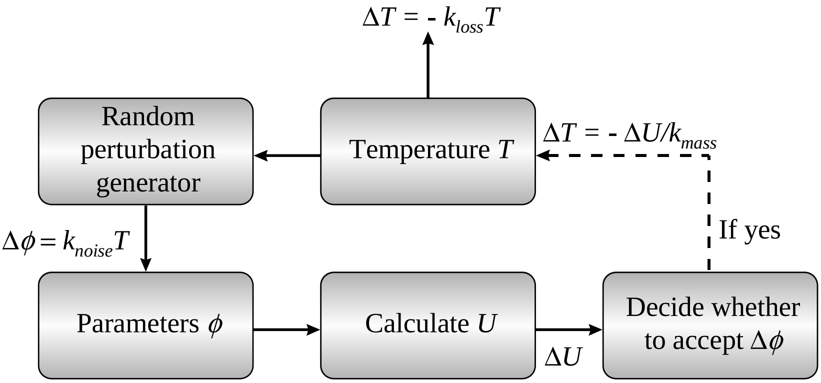

$\Delta T = - k_{loss} T$
Random perturbation generator
Temperature T
$\Delta T = - \Delta U / k_{mass}$
If yes
$\Delta\phi = k_{noise} T$
Parameters ϕ
Calculate U
ΔU
Decide whether to accept $\Delta\phi$

Figure 3.10 Our simulated annealing search algorithm.

Three parts have to be designed for the SA algorithm:

- Acceptance probability of worse solutions
- Random perturbation generator
- Temperature cooling schedule

For the acceptance probability we use

$$p_{accept} = e^{-\frac{\Delta U}{k_{boltz} T}} \tag{3.15}$$

where k_{boltz} plays a similar role as (but differs in value from) the Boltzmann constant from physics. This acceptance rule is generally used in almost all SA algorithms [Gema84]. For the random perturbation generator we a use a basic Gaussian random generator with three specific additional features. First, we use a Gaussian with zero mean and parameter specific variance given by:

$$\sigma_{\varphi_i} = k_{noise} T \Delta_i \tag{3.16}$$

The perturbations are regulated by temperature T, a scaling constant k_{noise} and parameter specific constants Δ_i, which represent the characteristic scale of each parameter, see Table 3.5. Whenever the prior σ of a parameter is zero, the corresponding perturbation scale parameter is set to zero.

Parameters	Δ	Parameters	Δ
b	1	$O_{PFL}^{z_{LFL}} \quad O_{PFR}^{z_{LFR}}$	100
$\varphi_{SF}^{LFL;x}$	10°	$O_{PFL}^{xy_{LFL}} \quad O_{PFR}^{xy_{LFR}}$	0.1
$\varphi_{SF}^{LFL;y} \quad \varphi_{SF}^{LFR;y}$	10°	$\varphi_{LFL}^{PFL;xy} \quad \varphi_{LFR}^{PFR;xy}$	0.1°
$\varphi_{SF}^{LFL;z} \quad \varphi_{SF}^{LFR;z}$	10°	$s_{y;L} \quad s_{y;R}$	1
$O_{CF_i}^{\sigma_{SF}}$	1	$K_{3;L}\ K_{5;L}\ K_{3;R}\ K_{5;R}$	0.1
$\varphi_{SF}^{CF_i;\sigma}$	10°	$\theta_L \quad \theta_R$	0.1°

Table 3.5 Parameter specific random perturbations.

Secondly, each parameter is only perturbed with a probability of 0.2. This enables the algorithm to continue when any of the scale sizes in Table 3.5 are (temporally) not appropriate. The third and final specific feature of the random generator is the generation of the average of all perturbations that provide better solutions:

$$\Delta\phi_{i+1}^{ave} = 0.99\Delta\phi_i^{ave} + 0.01\Delta\phi^{better} \tag{3.17}$$

After a while, the $\Delta\phi^{ave}$ will point more or less in the direction of the gradient towards better solutions. In each iteration, we randomly select either the Gaussian perturbation (3.16) or a perturbation that is a multiple of $\Delta\phi^{ave}$:

$$\Delta\phi = r\Delta\phi^{ave} \tag{3.18}$$

The r is a random number uniformly distributed between 0 and 2.2. Whenever $\Delta\phi^{ave}$ points in the right direction, the perturbation given by (3.18) produces better solutions, and thus the perturbation is fed back in (3.17). Since the average of r is slightly above 1, the scheme may show exponential convergence, severely increasing the speed of the SA algorithm. Whenever $\Delta\phi^{ave}$ points in the wrong direction or has the wrong scale, the scheme reduces to the normal Gaussian perturbations (3.16) until $\Delta\phi^{ave}$ again points in some right direction.

All the specific values mentioned above were determined experimentally on a heuristic basis.

In [Gema84], a temperature cooling schedule is derived for which the SA algorithm always yields the global minimum, or MAP solution. This schedule takes an infinite amount of time. For practical applications, we always have to cool faster. For this, no general rules are available. We will use an exponential scheme such as used in [Stil97], in which the temperature is lowered each step by:

$$\Delta T = -k_{loss} T \tag{3.19}$$

When the algorithm is working, now and then specific values for *T* are reached that result in random perturbations (3.16) of such a scale that many of them lead to a decrease in *U*. At these events, we do not want to lower the temperature, since we might miss the opportunity to lower *U* further. For this reason, we incorporated a feature that tries to keep the temperature constant at such events. It does this by giving a counter effect to the cooling (3.19), shown by the dotted line in Figure 3.10:

$$\Delta T = -k_{mass}^{-1} \Delta U \tag{3.20}$$

Whenever the solution is losing energy rapidly with a constant pace ($\Delta U << 0$), the temperature will remain fairly constant around an equilibrium given by (3.19) and (3.20).

The algorithm must be started with some initial temperature T_0 and solution ϕ_0 and ends when a minimum temperature T_{min} is reached. For ϕ_0 we take the mean value from the prior model in Table 3.4. Since *T* is always used together with scale factors k_{boltz}, k_{noise}, k_{loss} and k_{mass}, we can choose $T_0 = 1$ without loss of generality. The five remaining parameters are selected heuristically in the experimental section.

3.5.7 Evaluation of the search algorithm

Once the algorithm has stopped, we can verify if it has converged correctly. Using our prior knowledge about the marker detection algorithm we find the following expectation:

$$E\left[\left(m_i^{x_I} - m_{pred;i}^{x_I}(\phi)\right)^2\right] = E\left[\left(m_i^{y_I} - m_{pred;i}^{y_I}(\phi)\right)^2\right] = \frac{4N_{markers} - N_{params}}{4N_{markers}} \sigma_{mrk}^2 \tag{3.21}$$

The scale factor accounts for the fact that the estimated parameters will partly model the marker detection errors. The factor 4 originates from the fact that each point or marker is projected to both left and right images (factor 2) and produces an *x* and an *y* coordinate (factor 2). We now construct:

$$A^2 = \frac{1}{4N_{markers} - N_{params}} \sum_{\substack{\text{all markers} \\ \text{in all views} \\ \text{in both images}}} \left(m_i^{x_I} - m_{pred;i}^{x_I}(\phi)\right)^2 + \left(m_i^{y_I} - m_{pred;i}^{y_I}(\phi)\right)^2 \tag{3.22}$$

The *A* represents the total amount of marker position errors that are attributed to the marker detection algorithm, i.e. that could not be explained by the estimated specific camera model. We expect that *A* will be about σ_{mrk}. If this is more or less the case, we assume that

- the search algorithm has converged
- the marker detection algorithm works as expected
- the general camera model is appropriate

When A is too large, one of the above is not true. Additionally, this may be caused by a difference between the actual calibration object and its model. In [Pede99], this effect is actually used to refine the object model. When A is smaller, we assume that the marker detection algorithm works better than expected.

3.6 Fixed calibration experiments

We performed experiments with both synthetic and real images containing calibration objects. The results are evaluated with the following methods:

- Comparison of the remaining marker errors A with σ_{mrk} (2-D domain).
- Comparison of the model and reconstruction of the calibration object (3-D domain).
- Comparison of the estimated parameters with a ground truth (synthetic images only).

For the 3-D comparison we used both the single plate and the compound object from the N_{view} views of the plate. First, all markers in all views are reconstructed in 3-D space to yield the compound object. This is done by triangulating the marker positions in the left and the right image like a pair of corresponding points. The reconstructed object is then compared with the true compound object by calculating the differences in position, orientation and scale, and finally the remaining deformations. To reconstruct the plate model and compare it with the true model, the reconstructed compound object is split into the N_{view} parts. From each point (marker) the coordinates in its own *CF* frame are calculated. Then these coordinates are averaged over all views, which yields the reconstructed plate model. These errors are expressed in deformations in x, y and z directions.

3.6.1 Synthetic calibration object and images

We used a convergent camera setup with convergence angle $\alpha = 90°$, see Figure 3.11. First the simple camera model from Chapter 2, section 2.6 is used and then the general model from section 2.4, Table 2.3. The images have CCIR601 size N_x = 720 and N_y = 576. The calibration object consists of N_{view} = 3-4 views of the synthetic version of the A4 calibration plate. This plate is about 25 cm wide. To be sure that the compound object fits in the scene space, we set d to 35 cm in (2.57) and find $b\gamma \approx 0.5$. We set the focal length f arbitrarily to 1000, and then find with (2.51) that $\gamma \approx 0.6$. Thus, we select the baseline to be 80 cm.

The scene is then positioned 40 cm behind the cameras, according to (2.52). Via (2.56) we find that the scene resolution is about 0.7 mm. If the markers are estimated with 0.01 pixel accuracy, the triangulated scene points (marker centers) have a 3-D accuracy of 7 μm according to (2.67). This number is in general well below the 3-D accuracy of any geometrical model of practical calibration plates.

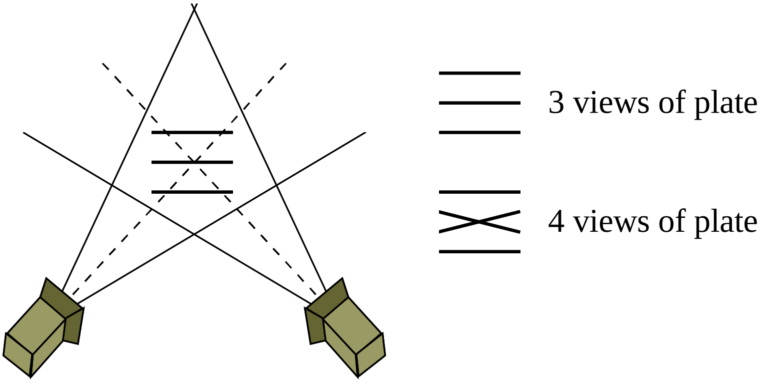


Figure 3.11 Setup for the stereo camera setup and the calibration object.

Table 3.6 shows the results of the fixed calibrations for the 3-view plate setup. As input for the fixed calibration scheme we used the known exact markers, the measured markers and markers of which the position contained deliberately noise of 1 pixel. For the exact markers we used $\sigma_{mrk} = 0.001$ to avoid infinity in (3.14). For the measured markers we used the $\sigma_{mrk} = 0.01$ found in section 3.4 (for this particular set of images we found $\sigma_{mrk} \approx 0.003$). The prior camera model is a pinhole model without CCD mispositioning/misorientation, skew or lens distortion. It contains the parameters shown in Table 3.6. The settings for the SA algorithm are shown in Table 3.7. The low value of k_{boltz} effectively inhibits any acceptance of worse solutions, i.e. only perturbations to better solutions are allowed. The SA algorithm showed the fastest convergence if the value of k_{mass} had been selected inversely proportional to σ^2_{mrk}. The computational load of the algorithm was between 1 to 20 minutes on an SGI octane computer.

From Table 3.6 we observe the following. In the experiment with the measured markers, the parameters are estimated very accurately. The errors in the plate reconstruction are in the order of the expected accuracy of 7 µm. The errors in the compound object are an order of magnitude larger. This is due to the errors in the estimated plate positions and orientations, which influence the reconstruction of the compound object. For the noised markers, the results are similar. The reconstruction accuracies for the plate and compound object are now only determined by the 1-pixel accuracy of the markers.

The experiment with the exact markers shows an example of how calibration with a compound object can go wrong catastrophically. The parameters are far off the true values, and the compound object is reconstructed with errors in the order of 2 cm. The plate reconstruction, however, is accurate up to 1 µm, and the marker positions are explained up to 0.001 pixel by the (false) parameters obtained. In a fixed calibration experiment with real objects and images, only the latter two quantities can be measured, and the very large error in the compound object remains unobservable.

Parameters	True	Estimated with exact markers $\sigma_{mrk} = 0$ (≈0.001)	Estimated with measured markers $\sigma_{mrk} = 0.01$	Estimated with noised markers $\sigma_{mrk} = 1$
baseline b	0.8	0.800	0.801	0.809
x-rotation $\varphi_{SF}^{LFL;x}$	0°	0.00°	0.00°	-0.06°
convergence $\varphi_{SF}^{LFL;y}$ $\varphi_{SF}^{LFR;y}$	-45° 45°	-56.2° 33.8°	-45.0° 45.0°	-44.8° 45.2°
z-rotation $\varphi_{SF}^{LFL;z}$ $\varphi_{SF}^{LFR;z}$	0° 0°	-0.00° 0.00°	0.00° 0.00°	0.15° 0.03°
focal lengths $O_{PFL}^{z_{LFL}}$ $O_{PFR}^{z_{LFR}}$	1000 1000	1492 670	1002 1000	1004 1017
pixel ratio $s_{y;L}$ $s_{y;R}$	1 1	1.270 0.999	1.000 0.999	0.994 1.002
Plate 0 position $O_{CF_0}^{\sigma_{SF}}$	0 0 -0.3	.114 .000 -.278	.000 .000 -.301	-.002 .000 -.305
Plate 0 orientation $\varphi_{CF_0}^{SF;\sigma}$	0° 0° 0°	.00° .00° .00°	.00° .00° .00°	.02° -.06° .01°
Plate 1 position $O_{CF_1}^{\sigma_{SF}}$	0 0 -0.4	.152 .000 -.370	.000 .000 -.401	-.002 .000 -.405
Plate 1 orientation $\varphi_{CF_1}^{SF;\sigma}$	0° 0° 0°	.00° .00° .00°	.00° .00° .00°	.01° -.01° .02°
Plate 2 position $O_{CF_2}^{\sigma_{SF}}$	0 0 -0.5	.190 .000 -.463	.001 .000 -.501	-.003 .000 -.505
Plate 2 orientation $\varphi_{CF_2}^{SF;\sigma}$	0° 0° 0°	.00° .00° .00°	.00° .00° .00°	-.24° -.18° .07°
SA alg. marker errors A	0	0.0010	0.0152	1.015
Compound reconstruction Position	0 0 0	.152 .000 .030	.000 .000 .001	.00 .00 .00
Orientation	0° 0° 0°	.00° -13.1° .00°	.00° -.04° .00°	-.00° .16° .04°
Scale	1	1.015	1.000	1.001
Deform	0	2 cm	62 µm	1 mm
Plate reconstruction σ_x σ_y σ_z	0 0 0	1 0 0 (µm)	6 6 8 (µm)	.5 .4 .6 (mm)

Table 3.6 Fixed calibration results with synthetic images and distortion-free cameras.

SA parameter	value
T_{min}	10^{-6}
k_{boltz}	10^{-6}
k_{noise}	10^{-3}
k_{loss}	10^{-4}
k_{mass}	10^{-8} to 10^{-2}

Table 3.7 Parameters for the SA algorithm with fixed calibration.

We repeated the three experiments several times and found that in all three cases, now and then the algorithm converges to a false solution. We also performed the experiments with the 4-view setup shown in Figure 3.11. This always yielded the right solution with similar accuracy as in Table 3.6. To be sure that this was no coincidence, we also performed several experiments where we forced one of the focal lengths to some wrong value. If the 3-plate setup was used, a solution could always be found with very low A. Thus, with false parameters (by definition in this experiment) the measured marker positions can be explained fully. The reliability of the parameters found can thus not be guaranteed nor evaluated by A, which is the only option when using real images. With the 4-plate setup, the value for A remained several orders higher than σ_{mrk}, indicating that it was no longer possible to estimate the parameters reliably. This is to be expected from a reliable algorithm, since one parameter was fixed to a wrong value. Thus, our fixed calibration scheme with the 4-plate setup yields reliable results whenever possible. Via A, this can be reliably verified.

We found that the prior model influences the results only due to the minima and maxima on the parameters. The μ and σ of the Gaussian part in the *TG* function has little effect. In the experiments in this section, the prior μ and the actual value of both focal length, pixel ratios and some angles are identical, which may seem to bias the results towards the right solution. However, during the run of the SA algorithm, many other focal lengths are encountered before converging to the right values. We repeated the experiments with different values for the prior model and it did not change the final results.

Table 3.8 shows experiments with a similar camera setup, but with the general camera model including lens distortion, CCD mispositioning, misorientation and skew. The four-plate setup was used. The columns give the true parameters, the parameters estimated with the measured markers using the general prior camera model from Table 3.4, and finally, the parameters estimated with the measured markers and a pinhole prior model.

We repeated these experiments extensively, but in none of them the algorithm converged completely to $A \approx \sigma_{mrk}$ after run times in the order of 30 minutes and more. Thus, the SA algorithm still limits the accuracy of the parameters in these experiments. However, the results show that all parameters for the undesired camera properties (lens distortion, CCD misposition and misorientation) can be measured. The accuracy varies widely from a few percent to only correctness of the sign. The experiment with the pinhole model gives an A of about 2.7, showing that the pinhole model cannot explain the observed markers well for cameras with distortions. The final reconstruction errors are an order of magnitude larger than those obtained using the general camera model.

Parameters	True	Estimated with measured markers σ_{mrk} = 0.01	Same, but with pinhole prior camera model
baseline b	0.8	.802	.803
x-rotation $\varphi_{SF}^{LFL;x}$	1°	0.608°	0.67°
convergence $\varphi_{SF}^{LFL;y}$ $\varphi_{SF}^{LFR;y}$	-45° 45°	-44.3° 44.6°	-45.93° 49.97°
z-rotation $\varphi_{SF}^{LFL;z}$ $\varphi_{SF}^{LFR;z}$	5° 5°	4.77° 5.25°	4.60° 4.90°
focal lengths $O_{PFL}^{z_{LFL}}$ $O_{PFR}^{z_{LFR}}$	800 900	802 900	813 884
pixel ratio $s_{y;L}$ $s_{y;R}$	1.1 0.9	1.088 0.895	1.160 0.958
Lens distortion $K_{3;L}$ $K_{3;R}$	1 -1	1.04 -1.04	0 0
$K_{5;L}$ $K_{5;R}$	0.5 -0.5	0.41 -0.39	0 0
CCD skew θ_L θ_R	0.1° -0.2°	0.06° -0.15°	0° 0°
mispositioning $O_{PFL}^{xy_{LFL}}$ $O_{PFR}^{xy_{LFR}}$	10 10 5 -5	2.8 0.3 4.1 -4.0	0 0 0 0
misorientation $\varphi_{LFL}^{PFL;xy}$ $\varphi_{LFR}^{PFR;xy}$	-4° -5° 5° 4°	-1.7° -7.5° 5.0° 4.9°	0° 0° 0° 0°
Plate 0 position $O_{CF_0}^{\sigma_{SF}}$	0 0 -0.3	.002 .001 -.305	-.018 -.002 -.269
Plate 0 orientation $\varphi_{CF_0}^{SF;\sigma}$	0° 0° 0°	.023° -.05° -.03°	-1.41° .15° -.06°
Plate 1 position $O_{CF_1}^{\sigma_{SF}}$	0 -0 -0.4	.002 .002 -.406	-.022 -.001 -.359
Plate 1 orientation $\varphi_{CF_1}^{SF;\sigma}$	0° 20° 0°	.22° 20.06° .10°	.015° 20.30° -.04°
Plate 2 position $O_{CF_2}^{\sigma_{SF}}$	0 0 -0.4	.002 .002 -.406	-.021 -.001 -.359
Plate 2 orientation $\varphi_{CF_2}^{SF;\sigma}$	0° -20° 0°	.23° -20.36° -.06°	.40° -15.81° -.09°
Plate 3 position $O_{CF_3}^{\sigma_{SF}}$	0 0 -0.5	.002 .002 -.507	-.025 -.002 -.449
Plate 3 orientation $\varphi_{CF_3}^{SF;\sigma}$	0° 0° 0°	.17° -.12° .06°	2.09° -.28° -.07°
SA alg. marker errors A	0	0.099	2.747
Compound reconstruction Position	0 0 0	.002 .002 -.006	-.0215 -.001 .042
Orientation	0° 0° 0°	.223° -.158° .014°	.12° 1.53° -.01°
Scale	1	1.004	0.9635
Deform	0	0.63 mm	6.7 mm
Plate reconstruction σ_x σ_y σ_z	0 0 0	65 46 39 (μm)	2.1 1.4 0.7 (mm)

Table 3.8 Fixed calibration results with synthetic images and the general camera model.

3.6.2 Real calibration object and images

For this experiment we used two semi-professional 3-CCD Panasonic WV-E550E cameras, equipped with Fujinon TV Zoom lenses. Their analog outputs were digitized to CCIR601 format with 720x576 pixels. Two setups were used; a slightly converging, almost parallel setup with wide-angle view (zoomed out) using the A1 calibration plate, and a more convergent setup with the A4 plate, similar to the setup used in the experiment with the synthetic data. In both cases, the 4-view compound object was used. The general camera model was used without skew and CCD misorientation as discussed in section 2.4, but including CCD mispositioning and lens distortion.

Parameters	Slightly converging setup A1 plate	More convergent setup A4 plate
baseline b	.440	.349
x-rotation $\varphi_{SF}^{LFL;x}$	-0.07°	-0.104°
convergence $\varphi_{SF}^{LFL;y}$ $\varphi_{SF}^{LFR;y}$	-7.37° 4.77°	-16.08° 15.83°
z-rotation $\varphi_{SF}^{LFL;z}$ $\varphi_{SF}^{LFR;z}$	-0.11° 0.22°	-0.773° 0.636°
focal lengths $O_{PFL}^{z_{LFL}}$ $O_{PFR}^{z_{LFR}}$	935 936	973 972
pixel ratio $s_{y;L}$ $s_{y;R}$	0.918 0.917	0.9124 0.9118
Lens distortion $K_{3;L}$ $K_{3;R}$	-0.208 -0.191	-0.171 -0.183
$K_{5;L}$ $K_{5;R}$	0.348 0.270	0.584 0.362
mispositioning $O_{PFL}^{xy_{LFL}}$	0.748 0.393	-5.682 -0.678
$O_{PFR}^{xy_{LFR}}$	0.295 0.753	-1.615 -1.011
Plate 0-1 z position $O_{CF_{0-1}}^{z_{SF}}$	-1.349 -1.581	-0.667 -0.770
Plate 2-3 z position $O_{CF_{2-3}}^{z_{SF}}$	-1.603 -1.948	-0.812 -0.951
SA alg. marker errors A	0.0765	0.0865
Plate reconstruction σ_x σ_y σ_z	.34 mm .21 mm .68 mm	88 µm 61 µm 115 µm

Table 3.9 Fixed calibration results with real cameras.

Table 3.9 shows representative results from several runs of the SA algorithm. From the plates only the z positions are shown. To assess the results, we included the following ground truth data. The baseline and plate positions were roughly measured manually. The accuracy of these measurements is limited, since our cameras have multiple lenses which causes the effective lens center to lie somewhere inside the camera housing. Further, the two cameras are of the same type. We expect that the pixel aspect ratios and the lens distortion parameters are the same for both cameras and both experiments. Finally, the zoom lenses were completely zoomed out in both experiments. We expect all four measured focal lengths to be approximately the same.

The parameters in the A1 experiment were in agreement with the measurements we carried out manually (up to a few cm). In the A4 experiment, the baseline was in agreement, but the z positions of the plate were about 10 cm further from the cameras than expected. We observe in both experiments that A is slightly below 0.1 pixel, from which we conclude that our marker detection scheme is at least accurate to 0.1 pixel for real images. Our expectations concerning the pixel aspect ratios, focal lengths and lens distortion parameter K_3 are more or less satisfied.

Since we used real images, no ground truth data was available for the compound scene reconstruction. The plate reconstruction could be compared with the plate model. To evaluate these results in Table 3.9, we first calculate the expected size and accuracy of the reconstructed scene. We approximate the two setups by the simple setup from section 2.6.1. For the convergence angle α we use the difference between the two φ_y angles and find $\alpha_{\mathrm{A1}} \approx 12°$ and $\alpha_{\mathrm{A4}} \approx 32°$. Then via (2.55) we find for the dimensions of the A1 scene about 1x1.5x10 m and for the A4 scene about 0.3x0.45x1 m. Filling in 0.1 (the observed marker detection error A) for σ_{cor} in (2.67), and $N \approx 650$ (the average number of pixels in horizontal and vertical direction) in (2.56), we find for the expected reconstruction accuracies for the A1 experiment 0.15x0.23x1.5 mm and for the A4 experiment 46x70x150 μm. These numbers have the same order of magnitude as the values obtained by reconstruction of the A1 and A4 plate. In both cases, the relative reconstruction accuracy is about 10^{-3} to 10^{-4}.

3.6.3 Discussion

Due to the wide variety in notation and types of assessment, it is usually quite hard to compare results in fixed calibration. We will compare our results with the classic result in [Tsai87] and the more recent results in [Wei94]. In [Tsai87] a planar calibration object of 2x1.5 inch is reconstructed up to about 1 mil = 0.001 inch. This yields a relative accuracy of about 10^{-3}. Our results are slightly better, which can be explained completely by the marker detection algorithm (0.3 to 0.5 pixel in [Tsai87] and < 0.1 pixel for our algorithm). In [Tsai87] it is found that a number of coplanar views of the calibration plate provides good results, while in our experiments we needed non-coplanar views. We have no theoretical explanation for this difference, but Table 3.6 provides an experimental proof that coplanar views are not always sufficient. In [Wei94] pixel accuracies of $A \approx 0.25$ pixel are reported, similar to our results. The absolute reconstruction errors for the calibration plate were in the order of 0.3 mm, however, without any reference to the size of the plate.

Summarizing all our experiments, we find:

- The accuracy of our marker detection scheme is better than 0.1 pixel for real images from real cameras.
- Fixed calibration can result in accurate camera parameters and subsequently, accurate acquisition of scenes. In our experiments with real-image data, our calibration plates

were reconstructed with relative accuracies of about 10^{-4}, comparable to other results in literature.

- The results have been obtained using the marker detection algorithm with an accuracy better than 0.1 pixel. For natural scene reconstructions, it can be assumed that the correspondences found in image pairs have an accuracy > 0.1 pixel. Then the scene reconstruction accuracy is limited by the accuracies of the correspondences, and not by camera calibration.
- Fixed calibration may lead to unreliable results, if a volumetric calibration object is composed by multiple views of a single, flat calibration plate. This effect can only be observed if accurate ground truth data of the plate positions and orientations is available, which is the case in our experiments with synthetic data. We found that the 3-plate and 4-plate setup depicted in Figure 3.11 provide unreliable and reliable results, respectively.

As fixed calibration is a mature area of research, it is not surprising that our calibration and plate reconstruction results compare but not outperform other results from literature. However, our algorithms are based on the Bayesian formulation with simulated annealing, which provides high flexibility in modeling the cameras (the prior model can be changed in an easy way), and most importantly, it makes the step towards the far less mature area of self-calibration very small.

3.7 Self-calibration

In self-calibration, the actual scene itself is used as calibration object instead of a special object. Figure 3.12 illustrates the principle of self-calibration, using a simple cube as scene. The calibration algorithm does not rely on the specific shape of the scene. First, corresponding pixels between the left and right image of the scene are estimated, which is discussed in Chapter 4 of this thesis for images of general scenes. The fact that the light rays of corresponding pixels must intersect, even if we do not know exactly at which position in space, provides a constraint on the camera parameters enabling their estimation. This corresponds to minimizing the intersection error $\left|V_{P_{SL}\text{ to }P_{SR}}\right|$ for all triangulated correspondences (see section 2.5).

All self-calibration approaches cannot recover the absolute scale of the scene, since no reference object is present (as in fixed calibration) that has units in meters. Only in special cases absolute scale can be recovered, e.g. by using special stereo cameras [Zane96] and a combination of fixed and self-calibration [Rede99d].

When lens distortion is not modeled (or not present), it can be proven that at most 7 camera parameters can be measured using self-calibration [Arms96, Csur97, Luon93]. Any camera model with more parameters results in uncomplete parameter estimation and thus can never provide geometrically correct scene reconstructions. In the worst case a projective reconstruction is the best we can obtain [Csur97], which yields distortions of both lengths and angles.

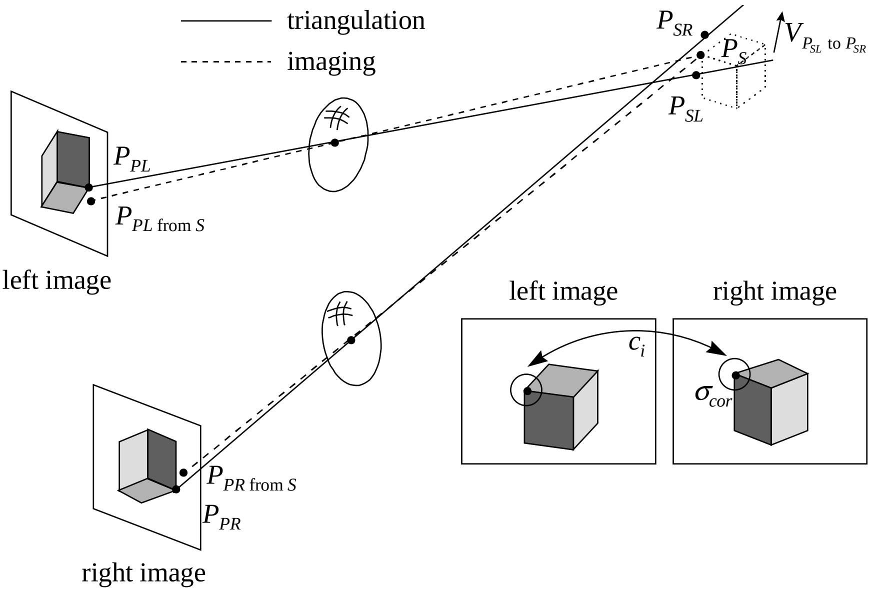


Figure 3.12 Self calibration principle. Each corresponding pixel pair P_{PL}, P_{PR} is triangulated to yield scene point P_S. For each such pair, the two light rays must intersect, which provides a constraint on the camera parameters enabling their estimation. Instead of actually minimizing the intersection error V in 3-D space, our algorithm projects P_S back to the images and calculates a similar error in the images.

For successful calibration in our application, we thus either need additional constraints or more information. Examples of extra constraints are fixing the CCD skew to zero and the pixel aspect ratio to 1 [Arms96, Boug98, Poll98]. More information is often obtained by taking more than two images into account, e.g. three images from a trinocular camera [Faug97, Fitz98], more than 3 images from a single camera [Jeba99] or two or more image pairs from a stereo camera [Deve96, Faug92, Luon93, Zhan93, Ziss95]. Unfortunately for this approach both the scene and the camera parameters must be static during the recordings.

The 7-parameter proof is derived under the assumption of ideal lenses without distortion. Any approach that first compensates for lens distortion and then performs self-calibration is able to measure all the distortion parameters plus the 7 aforementioned parameters. We will include lens distortion as an integral part of the self-calibration algorithm and show that this enables the measurement of additional camera parameters. In [Rede98d, Rede99b] our first results with this approach failed, due to an error in the implementation of the rotation matrices (A.12).

We will now discuss the choice of what to minimize (2-D or 3-D errors as in the fixed calibration case), the implementation in the Bayesian framework, the models used for the cameras and the correspondences, and the search algorithm. In all steps we will make full use of all results found for the fixed calibration scheme.

3.7.1 Minimize 2-D or 3-D measure

In section 3.5.1 we argued that both in fixed and self-calibration schemes it is wise to minimize errors in 2-D rather than 3-D errors. Figure 3.12 shows how we interpret the triangulation error in 2-D. First we triangulate the corresponding points P_{PL} and P_{PR} to construct the point P_S. During the self-calibration we project the point P_S back to the left and right and left images to obtain $P_{PL \text{ from } S}$ and $P_{PR \text{ from } S}$, respectively. The difference between P_{PL} and $P_{PL \text{ from } S}$ (and the difference between P_{PR} and $P_{PR \text{ from } S}$) is interpreted as an error from the correspondence estimator. For this error we have used a Gaussian probabilistic model with $\mu = 0$ and σ_{cor} due to the correspondence estimator.

Intuitively, an option for a difference measure in 3-D space is to minimize the triangulation intersection error $\left|V_{P_{SL} \text{ to } P_{SR}}\right|$. However, in this case the algorithm will have a bias towards small scene and will not pay attention to shape. If this is taken into account by making the measure invariant to scale, the method will start to resemble the 2-D method.

3.7.2 Bayesian formulation

In the self-calibration scheme we model the following parameters as random variables:

- Φ : all $N_{cam\text{-}model}$ parameters for the stereo camera model. The values of Φ are denoted by ϕ. The single parameters are ϕ_i.
- C : all N_{cor} estimated correspondences in the images. Correspondence estimators as discussed in the next chapter supply the values c of C. A single correspondence is denoted by $c_i = \{ P_{PL;i}^{xy_{IL}}, P_{PR;i}^{xy_{IR}} \}$.

Similar to the results found for the fixed calibration scheme in section 3.5.2, we obtain for the MAP solution:

$$\phi_{MAP} = \arg\max_{\phi} p_{C|\Phi}(c,\phi)\, p_{\Phi}(\phi) \qquad (3.23)$$

Here p_Φ is the prior camera model and $p_{C\,|\Phi}$ is the model for the errors in the estimated correspondences.

3.7.3 Prior camera model

We use the general prior camera model $p_\Phi(\phi)$ from Table 3.4, which was used for fixed calibration. However, we cannot measure any length in meters, but only in baselines as discussed in section 2.4.7, and thus use $\mu = 1$ and $\sigma = 0$ for the baseline b.

3.7.4 Correspondence model

In the correspondence model $p_{C|\Phi}(c,\phi)$, the differences between P_{PL} and $P_{PL\text{ from }SR}$ as well as P_{PR} and $P_{PR\text{ from }SL}$ are related to the errors from the correspondence estimator. This is similar to the fixed scheme in which $p_{M|\Phi}(m,\phi)$ is built from the differences between detected markers m and predicted markers m_{pred}.

The correspondence errors from the estimator are modeled as discussed in section 2.6.3, that is, both left and right points of an exact corresponding pair are detoriated by zero mean Gaussian noise with standard deviation σ_{cor}. Since we perform self-calibration *after* correspondence estimation, we cannot use epipolar geometry (see Figure 3.1). Then, the correspondence errors are isotropic in the x_I and y_I directions (see section 2.6.3). We then obtain:

$$p_{C|\Phi}(c,\phi) = \prod_{\substack{i \\ \text{all correspondences} \\ \text{in the stereo pair}}} \frac{1}{2\pi\sigma_{cor}^2} e^{-\frac{S_i}{2\sigma_{cor}^2}} \tag{3.24}$$

with

$$\begin{aligned} S_i = {} & \left(P_{PL;i}^{x_{IL}} - P_{PL\,\text{from}\,S;i}^{x_{IL}}(c,\phi)\right)^2 + \left(P_{PL;i}^{y_{IL}} - P_{PL\,\text{from}\,S;i}^{y_{IL}}(c,\phi)\right)^2 \\ & + \left(P_{PR;i}^{x_{IR}} - P_{PR\,\text{from}\,S;i}^{x_{IR}}(c,\phi)\right)^2 + \left(P_{PR;i}^{y_{IR}} - P_{PR\,\text{from}\,S;i}^{y_{IR}}(c,\phi)\right)^2 \end{aligned} \tag{3.25}$$

which is very similar to (3.10) for fixed calibration.

3.7.5 Search algorithm and its evaluation

We use the same Simulated Annealing algorithm as in the fixed calibration scheme, see section 3.5.6. We evaluate its convergence by A which is now defined by:

$$A^2 = \frac{1}{4N_{cor} - N_{params}} \sum_{\substack{i \\ \text{all} \\ \text{correspondences}}} S_i \approx \frac{1}{N_{cor}} \sum_{\substack{i \\ \text{all} \\ \text{correspondences}}} S_i \tag{3.26}$$

The A represents the total amount of errors that are attributed to the correspondence estimator, i.e. that could not be explained by the estimated specific camera model. The definition (3.26) follows the definition of A for fixed calibration (3.22). In self-calibration, N_{params} equals:

$$N_{params} = N_{cam-model} + 3N_{cor} \tag{3.27}$$

The $3N_{cor}$ equals the number of free parameters that the self-calibration procedure has when producing N_{cor} triangulated scene points (each has an x, y and a z coordinate). The simplification in (3.26) is only valid if $N_{cor} >> N_{cam\text{-}model}$.

We expect that $A \approx \sigma_{cor}$. If the calculated A is about σ_{cor}, we assume that

- the search algorithm has converged
- the correspondence estimator works as expected
- the camera model is appropriate

If A is too large, one of these is not true. In fixed calibration, A was also too large when the calibration object differed from the object model. In self-calibration, this is not possible since no object model is used. If cameras are calibrated with both methods and A is too large in the fixed method but as expected with self-calibration, we know that the calibration object differs from the object model. If A is smaller than expected, it must be due to a difference between the correspondence estimation errors and its model. The algorithm may work better than expected (lower σ_{cor}) or has errors different from the model (3.24) with independent zero mean Gaussians.

3.8 Self-calibration experiments

We will perform the same experiments as defined in section 3.6 for the fixed calibration scheme. We will use synthetic data and natural images. In both cases, our scene consists of several views of the (synthetic or real) calibration plate. This allows us to use the marker detection algorithm as correspondence estimator at this point, and thus $\sigma_{cor} = \sigma_{mrk}$. Although we use the calibration object as scene, the self-calibration method does not use the calibration object model.

We use exactly the same synthetic and real data as used for the fixed calibration scheme. Also the prior camera model is the same, with the exception of the baseline, which is set to 1 in self-calibration. Other specific exceptions are mentioned below. The simulated annealing algorithm was used without modification, working on a U function based on (3.24) instead of (3.10). The settings were identical to the settings for fixed calibration (Table 3.7) except for k_{noise}, the strength of the random perturbation generator. This was set to 0.1, instead of 10^{-3} for fixed calibration, which significantly increased the convergence speed in our experiments.

The results are evaluated with the following methods:

- Comparison of the remaining correspondence errors A with σ_{cor} (2-D domain).
- Comparison of the scene reconstruction with a ground truth (in the 3-D domain). This is possible only through the use of the calibration plate as scene.
- Comparison of the estimated parameters with a ground truth. For this we either use the true synthetic data or the parameters found by the fixed calibration scheme.

All results will be compared with the results obtained with fixed calibration.

3.8.1 Synthetic calibration plate and images

Table 3.10 shows the results from the self-calibration experiment with a pinhole stereo camera, similar to the fixed calibration experiment in Table 3.6. In all but one of the experiments, the pixel aspect ratio was fixed to 1 to set $N_{cam\text{-}model}$ to the theoretical maximum of 7. In one experiment, the pixel ratios of both cameras were added to the model, yielding $N_{cam\text{-}model} > 7$. According to the theory, in such a case reliable parameter estimation is not possible in self-calibration.

We found that the SA algorithm always converged quickly to a low value of A. Compared to the fixed calibration method, the run-times are smaller. Clearly, the fact that no plate positions and orientations have to be estimated outweighs the fact that four light rays have to be constructed instead of two in fixed calibration.

The value of A for the experiment with measured markers is much smaller than its value in the case of fixed calibration. In the other experiments, the As are the same for fixed and self-calibration. According to section 3.7.5, this can only be due to a difference between the actual correspondence estimation (marker localization) errors and the model (3.24) that assumes independent zero mean Gaussians. As found in section 3.4, the marker errors are concentrated on the outer ring, highly non-uniformly distributed over all 48 markers on the plate. If the errors in the outer ring are correlated between left and right views, this results in a reconstructed marker in 3-D space (see Figure 3.12) that has slightly moved from its correct position . In the fixed scheme, this would be noticed directly, since the inner markers are reconstructed well and both inner and outer markers must fit into the regular grid on the plate. The self-calibration scheme has no knowledge of the grid of the plate, which makes it insensitive for such left-right correlated errors on a few markers. Effectively, the model (3.24) is valid for fixed calibration, and for self-calibration if only the inner markers are used. We will use (3.24) also for self-calibration with all markers, since it is the best quantitative model we have at this moment.

Similar to the fixed calibration results, the self-calibration algorithm finds now and then a different solution that also explains the correspondences (markers in this case) with very low A. We observed this effect using the exact, measured and noised markers (the Table shows this effect for the true markers). In contrast to the fixed calibration scheme, we could not prevent this by using 4 views of the calibration plate. The self-calibration scheme has no knowledge of any calibration plate and, whatever the specific setup and number of views, deals with all measured markers as if they originate truly from one compound object. Thus, even with $N_{cam\text{-}model} = 7$, in our experiments, we were not able to provide reliable results with self-calibration. If one of the focal lengths was fixed to the true value, the algorithm always converged to the right solution.

Parameters		True	Estimated with exact markers $\sigma_{mrk} = 0$ (≈0.001)	Estimated with measured markers $\sigma_{mrk} = 0.01$	measured markers $\sigma_{mrk} = 0.01$, incl. pixel aspect ratios	Estimated with noised markers $\sigma_{mrk} = 1$
baseline	b	0.8	1	1	1	1
x-rotation	$\varphi_{SF}^{LFL;x}$	0°	0.00°	0.00°	0.00°	-0.04°
convergence	$\varphi_{SF}^{LFL;y}$ $\varphi_{SF}^{LFR;y}$	-45° 45°	-40.7° 40.7°	-44.8° 44.8°	-45.6° 39.3°	-35.7° 35.6°
z-rotation	$\varphi_{SF}^{LFL;z}$ $\varphi_{SF}^{LFR;z}$	0° 0°	0.00° 0.00°	0.00° 0.00°	0.00° 0.00°	0.13° 0.01°
focal lengths	$O_{PFL}^{z_{LFL}}$ $O_{PFR}^{z_{LFR}}$	1000 1000	859 859	994 994	1020 817	714 712
pixel ratio	$s_{y;L}$ $s_{y;R}$	1 1	1 1	1 1	1.52 1.35	1 1
SA alg. marker errors A		0	0.0010	0.0011	0.0012	1.061
Scene reconstruction	Position	0 0 0	.00 .00 -.187	.00 .00 -.103	.056 .00 -.147	.00 .00 -.318
	Orientation	0° 0° 0°	.00° .00° .00°	.00° .00° .00°	.0° -3.6° .0°	.0° .09° .04°
	Scale	1	1.617	1.262	1.604	2.345
	Deform	0	8.8 mm	0.34 mm	1.9 cm	2.0 cm

Table 3.10 Self-calibration results with distortion-free cameras.

From the scene reconstruction results we observe that the scene shifts forwards and backwards together with the scale difference, as discussed in section 2.6.4 (baseline section). It can be seen that the scale difference is not only caused by the inability of self-calibration to measure the baseline in meters (which yields a systematic scale difference of 1.25 in these experiments). Whenever a false solution is found that explains the observed marker positions just as well as the true solution, it also produces an extra scale difference, together with a larger reconstruction error. The deformation error is defined for the reconstructed scene if this scene is scaled to the size of the true scene (this only plays a role if the scale factor is different from 1).

The reconstruction results from the self-calibration method are about 5 times less accurate than those from the fixed method. This cannot be due to the accuracy of the correspondences (markers), since the same markers are used for triangulation in both experiments. Thus, the parameters obtained in self-calibration are less accurate than in fixed calibration. As expected, if we include the pixel aspect ratios in self-calibration ($N_{cam\text{-}model} > 7$), the errors on the parameters and the scene reconstruction increase drastically.

Table 3.11 shows the self-calibration results with the general camera model (compare with fixed calibration Table 3.8), with $N_{cam\text{-}model} = 23$. The calibration has been performed with the true and measured markers. From all our experiments, about one out of three showed convergence to A in the order of σ_{cor}. The table shows two representative experiments from this set. The other experiments converged to an A one or more orders higher than σ_{cor}.

Parameters	True	Estimated with exact markers, $\sigma_{cor} = 0$ (≈0.001)	Estimated with measured markers $\sigma_{cor} = 0.01$
baseline b	0.8	1	1
x-rotation $\varphi_{SF}^{LFL;x}$	1°	2.529°	1.409°
convergence $\varphi_{SF}^{LFL;y}$ $\varphi_{SF}^{LFR;y}$	-45° 45°	-41.40° 36.28°	-44.37° 41.56°
z-rotation $\varphi_{SF}^{LFL;z}$ $\varphi_{SF}^{LFR;z}$	5° 5°	4.30° 5.02°	4.90° 5.06°
focal lengths $O_{PFL}^{z_{LFL}}$ $O_{PFR}^{z_{LFR}}$	800 900	677 711	788 820
pixel ratio $s_{y;L}$ $s_{y;R}$	1.1 0.9	1.088 0.897	1.123 0.893
Lens distortion $K_{3;L}$ $K_{3;R}$	1 -1	0.72 -0.65	0.97 -0.84
$K_{5;L}$ $K_{5;R}$	0.5 -0.5	0.24 -0.14	0.48 -0.34
CCD skew θ_L θ_R	0.1° -0.2°	0.01° 0.37°	0.01° 0.01°
mispositioning $O_{PFL}^{xy_{LFL}}$	10 10	14.67 9.15	8.50 10.29
$O_{PFR}^{xy_{LFR}}$	5 -5	-7.15 -3.04	1.28 -4.39
misorientation $\varphi_{LFL}^{PFL;xy}$	-4° -5°	-3.67° -2.00°	-3.67° -2.00°
$\varphi_{LFR}^{PFR;xy}$	5° 4°	6.55° 8.58°	6.20° 5.76°
SA alg. marker errors A	0	0.0053	0.0053
Scene Translation	0 0 0	.051 .002 -.242	.028 .002 -.139
reconstruction Orientation	0° 0° 0°	.24° -3.77° .05°	.19° -1.62° .01°
Scale	1	1.815	1.378
Deform	0	1.3 cm	4.0 mm

Table 3.11 Self-calibration results with the general camera model.

Comparing these results with the self-calibration result in Table 3.10 (distortion-less cameras and 9-parameter model), we see a significant improvement. This clearly shows that the theoretical 7-parameter restriction for pinhole cameras does not apply in the same way to cameras with lens distortion. Comparing the results with Table 3.8 for fixed calibration, we notice that both fixed and self-calibration estimate all parameters with highly varying accuracy, and that fixed calibration provides slightly better results.

Unfortunately, we see that the false solutions as encountered in the 3-plate fixed calibrations and self-calibrations above are still present. The experiment with the exact markers shows reconstruction errors of 1.3 cm. As the dimensions of the scene are about 35x35x35 cm (see section 3.6.1), this still leads to a relative accuracy in the order of 10^{-1} to 10^{-2}.

3.8.2 Real calibration plate and images

Table 3.12 shows the self-calibration results with real cameras, obtained from the same data used for the fixed calibration experiment in Table 3.9. From the fact that A in self-calibration is as expected, while A in the fixed method is too large, we conclude according to section 3.7.5 that the inaccuracies in both our A1 and A4 plate models limit the accuracy of the fixed calibration scheme.

Parameters		Slightly converging setup, A1 plate	More convergent setup A4 plate
baseline	b	1	1
x-rotation	$\varphi_{SF}^{LFL;x}$	-0.004°	-0.219°
convergence	$\varphi_{SF}^{LFL;y}$ $\varphi_{SF}^{LFR;y}$	-8.24° 4.23°	-24.09° 22.34°
z-rotation	$\varphi_{SF}^{LFL;z}$ $\varphi_{SF}^{LFR;z}$	-0.08° 0.28°	-1.02° 0.75°
focal lengths	$O_{PFL}^{z_{LFL}}$ $O_{PFR}^{z_{LFR}}$	977 972	1435 1440
pixel ratio	$s_{y;L}$ $s_{y;R}$	1.033 1.030	1.350 1.368
Lens distortion	$K_{3;L}$ $K_{3;R}$	-0.24 -0.22	-0.36 -0.35
	$K_{5;L}$ $K_{5;R}$	0.40 0.27	1.04 1.06
mispositioning	$O_{PFL}^{xy_{LFL}}$	-14.92 -0.42	2.10 4.23
	$O_{PFR}^{xy_{LFR}}$	4.67 1.27	0.00 11.61
SA alg. marker errors	A	0.0330	0.0167
Scene reconstruction	Position	.073 -.082 -1.86	.016 .013 -.523
	Orientation	1.08° -.58° .06°	-.14° 1.47° .36°
	Scale	2.088	1.146
	Deform	2.4 cm	5.0 cm

Table 3.12 Self-calibration results with real cameras.

Further, we observe that the parameters in Table 3.12 do not resemble those of Table 3.9 very well. In this sense, the cameras have not been calibrated well. Scene reconstruction errors were calculated using the compound scene reconstruction results from fixed calibration as ground truth data. Then, the relative errors of the self-calibration scheme with respect to the fixed scheme are about 10^{-2} (A1 experiment, 2.4 cm error in a scene of 1x1.5x10 meter) and 10^{-1} (A4 experiment, 5.0 cm in a scene of 0.3x0.45x1 meter).

3.8.3 Discussion

We will compare our results with other results from literature. Most self-calibration approaches use more information than our scheme (more than two images, more than two cameras, more prior information), and reconstruction errors are seldomly given (due to the absence of a well-defined length unit). Therefore we selected to review a few more or less comparable results.

In [Pede97a] and [Pede99] an approach is adopted that stands midway between a fixed and a self-calibration method. A rough calibration object model is used (actual self-calibration does not use any object model at all). Pixel accuracies are reported of $A \approx 0.2$ and reconstruction accuracies of about 0.2 to 0.4 mm (absolute scale can be recovered by this method). The latter is obtained for real images, and refers to the reconstruction of the calibration object. As shown by our experiments with synthetic images, this accuracy may not hold for the compound object or scene. In [Zhan93], two image pairs are used (four images) and a camera model without lens distortion. Reconstruction results are reported of 10^{-2}, but their approach suffered from convergence (stability) problems. In [Arms96, Azar95, Poll98] some comparable results are reported for the structure-from-motion applications, where a single camera takes N images from a static scene at different angles. In these approaches, no lens distortion is considered. In [Arms96], an image sequence of 24 frames is used in which 128 scene points are tracked. Pixel errors (A) of about 0.1 pixel are reported, and relative accuracies for focal lengths and pixel aspect ratios of about 0.1%, and for skew and CCD mispositioning of about 10-25%. In [Azar95], 20 to 40 images are used and 5% errors in the focal length are reported. In [Poll98] a number of images in the order of 10 are used to reconstruct real scenes. Prior knowledge of e.g. the pixel aspect ratio is required. Angles between parallel and orthogonal lines in the scene are estimated with accuracies of about 1°, which cannot be compared easily with our results. Relative accuracies for lengths are given in the order of 10^{-1} to 10^{-2}. This is comparable to our results obtained with only a single stereo image pair and less prior camera knowledge, but it must be noted that their scene is more complex than ours.

From all the self-calibration experiments we conclude that:

- The theoretical restriction $N_{cam\text{-}model} \leq 7$ does not hold for self-calibration approaches that take into account lens distortion.
- Self-calibration of stereo cameras in a typical setup may result in a solution that fully explains the observed correspondences, but does not reconstruct the scene well. This holds even with correspondences up to 0.001 pixels. This was found both for cameras without lens distortion and a camera model with $N_{cam\text{-}model} = 7$, as well as for cameras with lens distortion and $N_{cam\text{-}model} > 7$, both using synthetic and real images.
- In our experiments, we observed relative accuracies for the scene reconstruction that range from 10^{-3} (synthetic scenes/images, pinhole camera model) to 10^{-2} and as low as 10^{-1} (real scenes/images, general camera model).
- Our self-calibration approach obtains results similar to other approaches in literature. However, we use the minimum of two images and little prior camera knowledge. Other

approaches use a higher number of images and cameras, and need more prior knowledge of the cameras and/or the scene.

3.9 Conclusions

We examined the calibration of a stereo camera. Two methods were investigated; fixed calibration and self-calibration. Fixed calibration has been used successfully for many years. It involves placing a special calibration object in front of the cameras and recording it, which requires the object and user interaction. Current research is focusing more and more on self-calibration that uses the actual scene as object, enabling fully automatic calibration.

Our contributions are as follows. First, we used the Bayesian probability framework to allow for a unified approach for both fixed and self-calibration methods. Further, we applied simulated annealing as general search algorithm in both calibration methods. With this algorithm we circumvent much analytic work, such as the computation of derivatives in gradient-based algorithms, or an initial approximation of the solution. The Bayesian approach together with simulated annealing provides high flexibility in modeling the cameras. Changes can be made directly via a prior camera model (fixing parameters to prescribed or hand-measured values), while the SA algorithm does not need any adaptation. Simulated annealing is computationally intensive; a single calibration took up to 30 minutes on a modern computer system. Future research may focus on speeding up this algorithm, the use of other algorithms or analytic approaches that provide (part of) direct solutions.

For the fixed calibration method, we followed an approach used in many current schemes, in which a virtual calibration object is composed by several views of a single planar calibration plate. We designed a new scheme for detecting markers on such plates within the images. The scheme is fully automatic and very robust. In synthetic images including noise and other practical image detoriations, we obtained positional accuracies of 0.01 pixel. In real images the results were experimentally shown to be better than 0.1 pixel. These results outperform most of the current algorithms for marker detection. Detailed research directions for further improvement have been given. The most prominent is to circumvent the errors in the markers at the perimeter of the plate, which are much larger than the errors in the other markers.

Our results with fixed calibration showed that real calibration plates could be reconstructed with relative accuracies up to 10^{-4}. Using synthetic data where the ground truth was available, we observed that the compound object, consisting of several views of the plate, could not always be reconstructed reliably even if marker positions were accurate up to 0.001 pixel. This also holds for the reconstruction of the actual scene of interest. The effect cannot be noticed in experiments with real calibration plates and images, and thus may lead to pseudo-accurate reconstructions. When we chose a set of 4 views of the calibration plate in a specific non-parallel orientations, we found that the reconstructions are always reliable. A direction for future research is the thorough investigation of the requirements on the setup for reliable calibrations.

In self-calibration, we considered the most difficult case that estimates all camera parameters using only a single image pair. For camera models without lens distortion, a theoretical proof exists saying that at most 7 parameters can be measured. We provided experimental evidence that this proof does not apply to cameras with lens distortion. Using synthetic data, we could measure more than seven parameters, with varying accuracy. A direction for research would be to prove this conjecture theoretically. Unfortunately, the non-linear aspects of lens distortion obscure such an analysis. Even projective geometry, a powerful and often used mathematical tool in self-calibration of lens-distortion-free cameras, cannot integrate lens distortion easily.

In our experiments, the unreliability encountered in fixed calibration appeared similarly in self-calibration, both in the cases with and without lens distortion (even with the theoretical maximum of 7 parameters). The solution in fixed calibration, the specific setup of a number of calibration plates, cannot be used by the self-calibration method since it does not use the calibration plate model. We did not find any solution for this problem. Still, even in these situations, we could calibrate the cameras such that scenes could be reconstructed with relative accuracies of about 10^{-1} to 10^{-2}. These results are comparable to other results found in literature, while those approaches use more than two images and more prior knowledge about the cameras and the scene. A heading for future research is how to ensure the reliability of self-calibration. Combining our self-calibration approach with multi-camera systems may provide a solution.

Chapter 4

Correspondence estimation

4.1 Introduction

The goal of this chapter is to derive a correspondence estimator that is especially suited for our 3-D scene acquisition application. Figure 4.1 shows the place of correspondence estimation (CE) in the scene acquisition process. The CE step is the most complex part of the process of 3-D scene acquisition from stereo images.

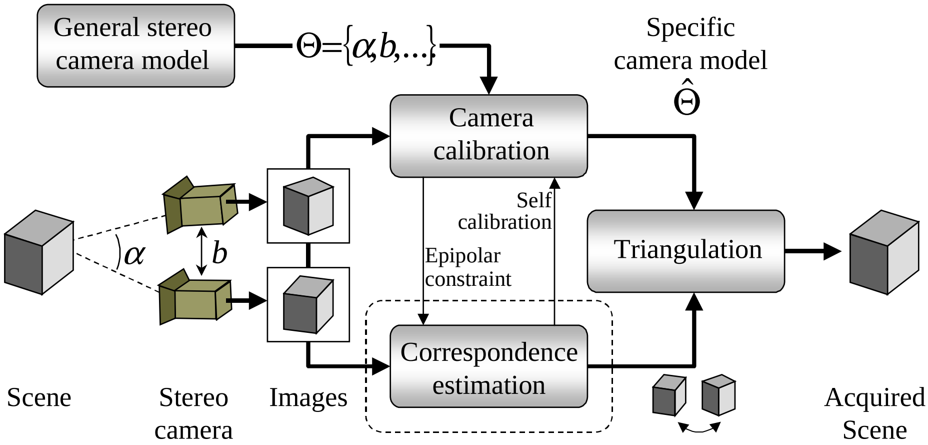


Figure 4.1 The scene acquisition process. This chapter deals with the correspondence estimation algorithm (shown dotted).

In literature, the topic of correspondence estimation has received already much attention because it plays a vital role in many other applications too. These include video coding, frame rate conversion, multi-viewpoint image generation, camera calibration and structure from motion. The requirements on the correspondences differs strongly per application, see Figure 4.2.

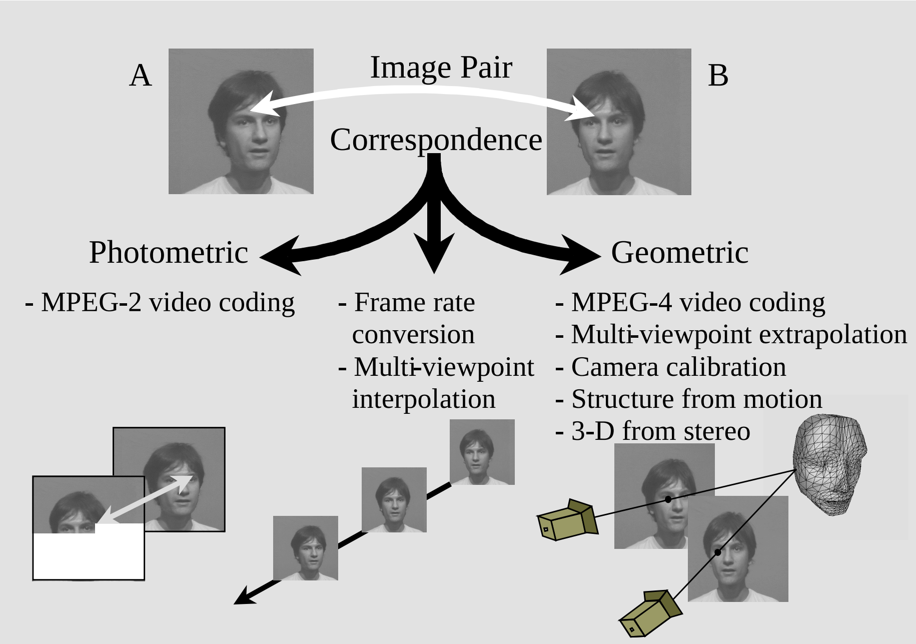


Figure 4.2 Applications of correspondence estimation. Some require photometric correspondences, while the majority of modern applications require geometric correspondences.

In video coding such as MPEG-2, correspondences are motion vectors from one image to the next in a sequence. The luminance of each pixel in an image is copied from or predicted by the previous image along a motion vector. Together the vectors are called the motion vector field, or the correspondence field. In the coding application, the correspondences have a photometric meaning. In many other applications, correspondences are used to extract 3-D information of scene points, giving them a geometric meaning. Such applications are camera calibration (see Chapters 2 and 3), structure from motion [Azar95, Matt89, Poll98], multi-viewpoint extrapolation [Rede97b] and 3-D from stereo [Rede99c]. Frame-rate conversion [Haan92] and multi-viewpoint image interpolation [Ohm98, Tsen95] lie more or less in between the photometric and geometric extremes.

For our 3-D scene acquisition application, we require high-resolution (pixel-dense) and high-accuracy geometric correspondence fields. Further, these fields must be calculated in real-time for dynamic scenes. This is in contrast with e.g. camera calibration, which can be performed once if the camera setup is static. Thus, for any real-time implementation in an actual 3-D communication system, we require that the computational load of the correspondence estimation is sufficiently low. At this moment, estimators do not exist that fulfil both requirements. This is due to the following reasons.

The high-resolution and high-accuracy estimation of geometric correspondences requires complex prior models for the dense field. These reflect the prior model of the scene to be acquired. Basically a smoothness constraint is imposed on the field, or equivalently, the scene. Such a model can only be designed on a heuristic basis, which is a reason for the large diversity in CE algorithms.

As a model does not have causal properties (along the image axes) in general, the estimation of all correspondences should in principle be done at the same time. If the correspondences for the entire image are estimated one by one, a causality constraint is implicitly imposed on the dependencies between the correspondences in the model. This may enable a fast implementation, but it restricts the accuracy.

The simultaneous and accurate estimation of all geometric correspondences in a pixel-dense field is a challenge for two reasons [Konr92]: First, the dimensionality of the solution space is extremely large: in the order of 10^6, the number of pixels in the image. This is computationally very demanding and may result in hours of computing per stereo image pair. Even with the continuing increase in computational power, new algorithms must be found in order to allow for real-time implementations in the near future.

Secondly, the estimation of a geometric correspondence field on the basis of photometric luminance fields of an image pair is not straightforward. It is completely based on heuristics as mentioned. The most commonly used heuristic is the so-called Constant Image Brightness (CIB) assumption [Horn86]. It states that a corresponding pixel pair has equal luminance. In Figure 4.3, two contours of equal luminance are depicted in an image pair. If we take a point P_A on the contour in image A, the question is to which point in image B it corresponds.

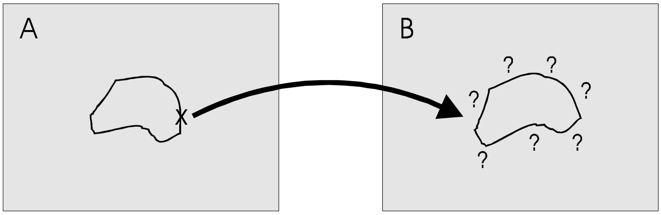


Figure 4.3 Photometric similarity is insufficient in geometric correspondence estimation.

For a photometric correspondence, all points on the contour in B would do. But there is only one point that corresponds geometrically, and we cannot be sure whether it lies on the contour in B, or not. If it does not, this can be due to, for example, camera noise, specular reflectivity of scene surfaces or to the use of a stereo camera with unbalanced photometric properties. Thus, we cannot use the CIB constraint alone to estimate a dense field of geometric correspondences. For this reason correspondence estimation is often called an ill-posed problem [Bert88]. Additional geometric constraints are needed, together with an appropriate photometric model that accounts for deviations from the CIB model.

In order to meet our goal in this challenging field, this chapter contains two parts. In the first part from sections 4.2 to 4.8, we give a thorough overview of classic and modern techniques for correspondence estimation. We will evaluate these using the requirements of our application. The overview is based on [Rede99a]. In the second part in section 4.9, we will use the review for the design of two new estimators for our application. One is suited for calibrated cameras, that uses the epipolar constraint (see Appendix B). The second algorithm is suited for uncalibrated cameras, that works autonomously and can be used for self-calibration purposes. Figure 4.1 indicates these two options.

First we will formally define geometric correspondence and investigate the different types of image pairs in section 4.2. Then in section 4.3 we look briefly at the classic approaches to correspondence estimation and at their feasibility and flaws for simultaneous dense estimation. In section 4.4 we will focus on the Bayesian approach, which is very suitable for this task and for which several promising algorithms have recently been developed. This approach utilizes four distinct steps, which are treated in sections 4.5 to 4.8. In section 4.9 the two new estimators are proposed. Finally, section 4.10 concludes the chapter.

4.2 Geometric correspondence in image pairs

We will first formally define geometric correspondence in section 4.2.1. Up to this point, the image pairs in all applications can be categorized in three types: spatial, parallel and temporal image pairs. In section 4.2.2 we will discuss spatial image pairs, which are provided by stereo cameras, as in our application. Section 4.2.3 describes parallel image pairs, which are a special type of spatial image pairs. Section 4.2.4 deals with temporal image pairs that originate from a single camera taking multiple images in a sequence. Although this is not directly related to our application, it is the application for which the majority of CE algorithms have been derived.

4.2.1 Definition of geometric correspondence

If the luminance of a point P_A in image A and a point P_B in image B are defined by the same scene point, we say that P_A and P_B correspond (see Figure 4.4). From this point on, we will mean geometric correspondence whenever we mention correspondence, unless stated otherwise.

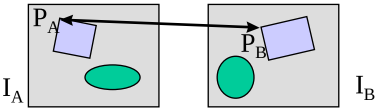


Figure 4.4 Correspondence between P_A and P_B.

Due to object transparency and camera defocus, the luminance of one point in an image may be defined by several scene points at the same time. This holds for both images, giving rise to multiple (many-to-many) correspondences (see Figure 4.5).

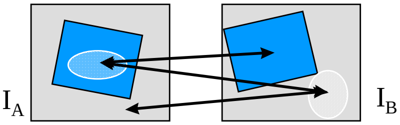


Figure 4.5 Multiple correspondences.

We have not discovered in literature any attempt to take multiple correspondences into account in the simultaneous estimation of dense correspondence fields. Therefore, from this point on, we will assume that all scene objects are opaque (not transparent).

Opaque objects that move in front of each other cause occlusion in images. It is possible that a scene point P is visible in image A as P_A, while in image B it is occluded by another scene point, Q, visible in B as Q_B. We define that there is a pseudo-correspondence from P_A to Q_B (see Figure 4.6). The point P_A is called an occlusion point.

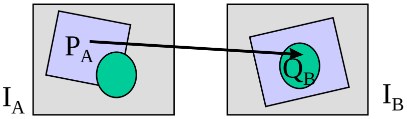


Figure 4.6 Pseudo-correspondence from occlusion point P_A to some point Q_B.

Pseudo-correspondences enhance the quality of images generated in multi-viewpoint and frame-rate conversion applications. They provide information about the position of point P in all intermediate images in which P is visible. In the applications 3-D from stereo and structure from motion, the models obtained are more complete as the pseudo-correspondences can be used to extract additional 3-D scene points. It is expected, however, that pseudo-correspondences can be obtained less accurately than real correspondences, since no photometric constraints are available for their estimation. Geometric constraints are the only clue. We expect e.g. that the pseudo-correspondence originating at P_A is similar to real correspondences obtained in the vicinity of P_A.

4.2.2 Spatial image pairs

Spatial image pairs are obtained when a scene is recorded by two cameras, A and B, which are located at different positions (see Figure 4.7). This setup is used in our application (where a sequence of such pairs is recorded).

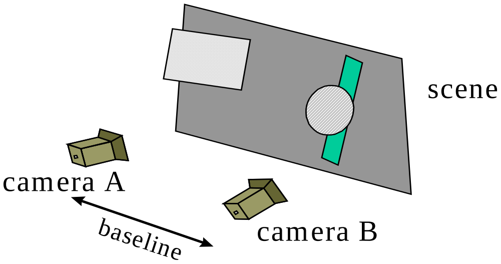


Figure 4.7 A stereo camera provides a spatial image pair.

For spatial image pairs, epipolar geometry provides a very powerful restriction on correspondences that has general validity, see Appendix B. If two points from an image pair correspond, they should lie on conjugate epipolar lines. This is called the epipolar constraint. It reduces the set of possible correspondence candidates for a point in image A

from all points in image *B* to only those on the conjugated epipolar line in *B* (see Figure 4.8).

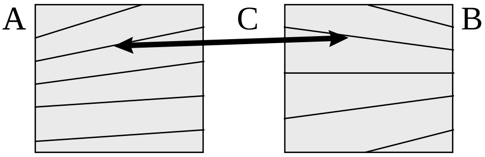


Figure 4.8 Correspondence is restricted to conjugate epipolar lines.

This restriction on the correspondences reduces the complexity of estimation by an order of magnitude, since every pixel in image *A* now only has a 1-dimensional set of pixels in image *B* as potential correspondences, in stead of a 2-dimensional set. If the epipolar constraint is used, correspondence estimation is called disparity estimation. Epipolar geometry can only be used if the cameras are already calibrated (see Figure 4.1).

The distance between the cameras is called the baseline. The larger the baseline, the more accurate the triangulation given the finite accuracy of the estimated correspondences, see Chapter 2. However, large baselines also yield large differences in the image pair (see Figure 4.9), which is a challenge for the estimation algorithms.

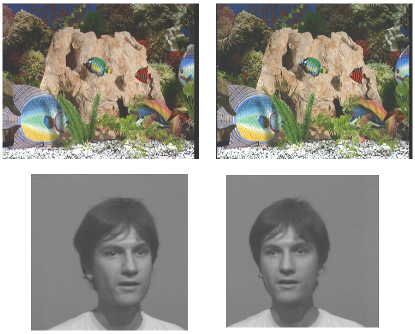


Figure 4.9 Small (top) and large (bottom) baseline.

4.2.3 Parallel image pairs

A special type of spatial image pairs arises if the cameras are in the parallel setup discussed in section 2.4.9. This setup requires that two identical pinhole cameras (no lens distortion or CCD misplacement) are placed with equal orientations, while their positions differ only in the direction of the scan lines. In this way a parallel image pair is obtained, in which corresponding pixel pairs lie on equal scan lines (see Figure 4.10).

In a parallel image pair the scan lines coincide with the epipolar lines, see Appendix B. The epipolar constraint is then applied by simply removing the y components from a correspondence field, which is then called a disparity field. Since the search for correspondences is now restrained to a horizontal range only, disparity estimation is much less complex than correspondence estimation. Disparity estimation algorithms for parallel image pairs are widely available [Cox96, Fran96, Inti94, Rede98a, Tsen95, Woo96].

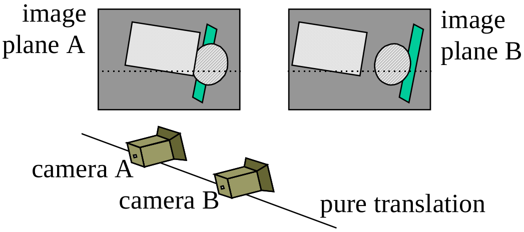


Figure 4.10 A parallel camera setup provides a parallel image pair in which corresponding pixel pairs always share the same scan line. Correspondence estimation is reduced to the much less complex disparity estimation, requiring only a horizontal search.

When cameras providing a spatial image pair are calibrated, the *A* and *B* images can be rectified as discussed in section 2.4.9. The result is a parallel image pair *A'*, *B'* in which disparity can be estimated by means of algorithms for parallel pairs.

4.2.4 Temporal image pairs

Temporal image pairs are obtained by recording a scene by a single camera that takes a shot at two different time instants, t_A and t_B (see Figure 4.11).

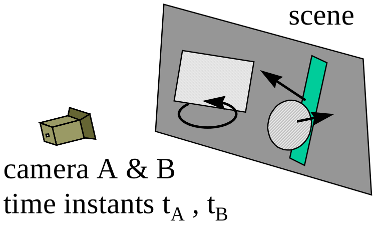


Figure 4.11 A single camera provides a temporal image pair.

The correspondences are related to the motion of scene objects. Correspondence estimation in temporal image pairs is therefore called motion estimation [Chan94, Konr92, Stil97].

For scenes with several rigid objects moving independently, temporal and spatial image pairs can be converted into one another, on an object-by-object basis. This enables the use of the strong epipolar constraint also in temporal image pairs [Xu96].

Figure 4.12 shows a scene with a number of rigid objects moving differently, recorded by a single camera. Figure 4.13 shows the circular object from the scene, recorded by two virtual spatial cameras. The resultant temporal and spatial image pairs are the same as far as the part of the circular object is concerned. The difference in the positions and orientations of the virtual *A'* and *B'* cameras relate to the translation and rotation of the circular object. Obviously, the epipolar constraint can be applied on the spatial image pair in Figure 4.13.

As the temporal pair is the same within the circular object, the exact same constraint holds. Figure 4.12 shows the epipolar geometry for all objects.

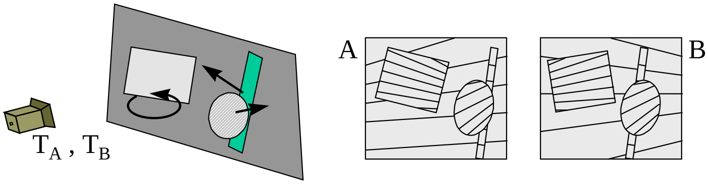


Figure 4.12 A camera records a temporal image pair from a scene with several rigid objects moving independently. Each of the objects has its own epipolar geometry.

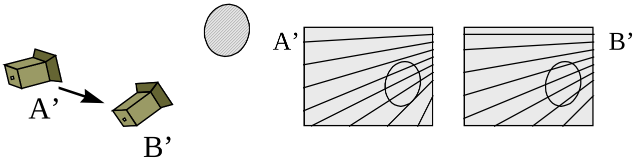


Figure 4.13 Spatial construction of one rigid object, selected from a temporal image pair.

In structure from motion applications, the 3-D triangulation of an object can be done in the same way as in our 3-D from stereo application, with one exception. The two virtual spatial cameras cannot be calibrated off line by fixed calibration methods. Self-calibration techniques have to be used on the basis of the estimated correspondence field. If the scene consists of only one object, the spatial and temporal image pairs are the same, which is used in [Rede98d, Poll98].

4.3 Classic correspondence estimation methods

We will briefly discuss the classic approaches to correspondence estimation: feature detection and matching, block matching, pel-recursive and optical-flow techniques. For more details we refer the reader to the excellent overview in [Teka95a].

4.3.1 Feature-based algorithms

Feature-based algorithms [Barn80, Liu93] first extract predefined features, and then match these (see Figure 4.14). The separation of detection and matching is a restriction on the quality that can be obtained.

The definition of features is not easy. The most well-known and general feature is an edge, of which definition and estimation has been investigated over long periods [Boye94, Canny86]. This approach yields a sparse correspondence field.

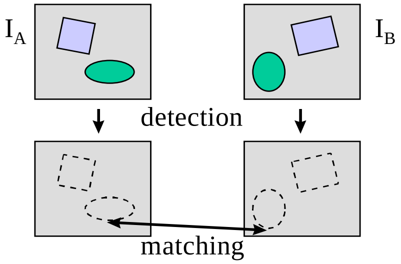


Figure 4.14 Feature detection and matching.

4.3.2 Block-matching algorithms

In block matching, rectangular blocks of pixels are matched [Acca95, Hend96] (see Figure 4.15). For each block in image *B*, a block is sought in image *A* which most resembles the block in *B* according to some criterion. A dense field can be obtained by means of interpolation or the use of overlapping blocks.

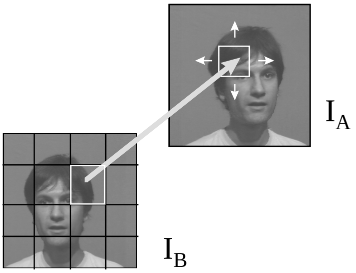


Figure 4.15 Block matching.

During estimation, a single correspondence vector is used for all pixels within one block. Since the vector only models translation, this approach does not work well for rotated and skewed objects in an image pair.

For large textured areas undergoing relatively uniform motion, large blocks enable high-accuracy estimation of correspondences. The uniform motion restriction, however, limits the resolution obtained. To some extent, this can be overcome by adapting the block size to the image content [Kana94].

4.3.3 Pel-recursive algorithms

These algorithms [Biem87, Börö91] have been developed for image-sequence coding. They obtain a dense field by scanning, i.e. they start the estimation at the upper-left pixel and end at the bottom-right pixel (see Figure 4.16).

First, the luminance of pixel *x* in image *B* is predicted from image *A* by means of the correspondence vector found at the previous pixel in *B* (pixel 6 in Figure 4.16). Then a group of *N* pixels (here *N* = 7) is matched to image *A*. The group has a ‘causal’ shape in the sense that it contains only pixels with known luminance in *B*. In the pel-recursive approach an analytical expression is used to obtain the new vector on the basis of the previous one. It is assumed that the previous vector is a good estimate of the new vector and thus, only small changes are allowed between two vectors.

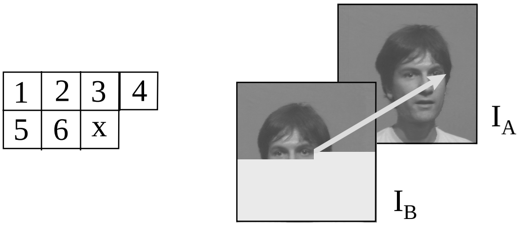


Figure 4.16 Pel-recursive technique.

The regular structure and causality of block matching and pel-recursive techniques make it possible to implement them efficiently in hardware [Hend96, Pano98a]. However, the causality restricts the quality of the correspondences obtained.

4.3.4 Optical-flow algorithms

This method is the first approach to the simultaneous estimation of a dense correspondence field [Horn86]. The method uses the following relation between photometric correspondence vectors with components (C_x,C_y), and the spatial and temporal derivatives of luminance in an image sequence:

$$\left(C_x \cdot \frac{\partial}{\partial x} + C_y \cdot \frac{\partial}{\partial y} + \frac{\partial}{\partial t}\right) I(x, y, t) = 0 \tag{4.1}$$

An additional regularization term biases the solution towards a globally smooth correspondence field [Horn86, Tsai97]. Discontinuity fields have been incorporated to avoid oversmoothing at object boundaries [Heit93].

The drawback of this approach is that the luminance derivatives are approximated numerically. This requires local linearity of luminance in both spatial and temporal directions. In image sequences with large motion (fast moving objects) the local linearity is violated. In stereo applications, the temporal axis is replaced by a camera position axis. For a camera baseline of any reasonable size, the position linearity is violated.

4.4 Bayesian correspondence estimation

More recent approaches to correspondence estimation are the Bayesian methods, which are applied both to temporal image pairs [Chan94, Konr92, Stil97, Teka95b, Zhan93] and to spatial pairs [Cox96, Rede98d, Woo96]. In this approach the simultaneous estimation of dense correspondence fields is possible. The luminance derivatives in the optical flow method are avoided. Further, the estimation of other information besides correspondences (e.g. object segmentation) can be incorporated.

In the Bayesian approach or framework we distinguish four steps, depicted in Figure 4.17.

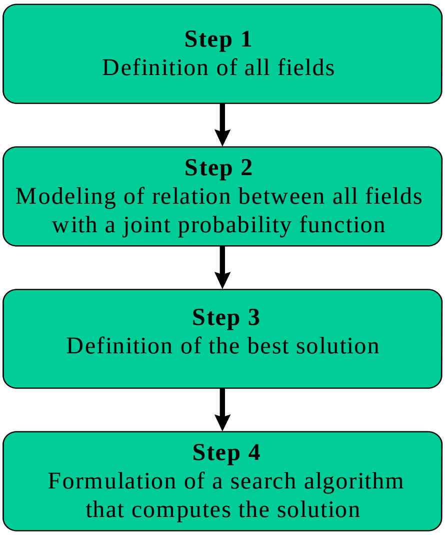

Step 1
Definition of all fields
Step 2
Modeling of relation between all fields with a joint probability function
Step 3
Definition of the best solution
Step 4
Formulation of a search algorithm that computes the solution

Figure 4.17 The Bayesian framework.

The separation of problem statement in the first three steps and the derivation of a search algorithm in step 4 [Drie92] increases the portability and adaptability of algorithms among different applications and different designers.

In the first step, we define the input images I_A and I_B and all output fields $\{F_1, F_2, ...\}$ to be estimated. The output fields represent correspondence, occlusion and possibly discontinuity and segmentation fields. In step 2 the relations between all these fields are modeled with a joint probability function in $F = \{F_1, F_2, ...\}$, conditioned by the observed image pair i_A, i_B:

$$p_{F|I_A,I_B}\left(f, i_A, i_B\right) \tag{4.2}$$

This is a density in the continuous fields in F and a mass function in the discrete fields. In the remainder of the chapter we will not refer to this explicitly. The design of the joint model is usually decomposed by means of the Bayes rule, which accounts for the name of these approaches:

$$p_{F_1,F_2,F_3|I_A,I_B} = p_{F_3|F_1,F_2,I_A,I_B}\, p_{F_2|F_1,I_A,I_B} \cdot p_{F_1|I_A,I_B} \tag{4.3}$$

In the third step the best solution f_{SOL} is defined by a criterion on the probability function, such as the Maximum A Posteriori (MAP) criterion. In the fourth and final step a search algorithm is formulated that computes the defined solution or a relevant approximation.

We will now focus on each of the four steps in the Bayesian framework.

4.5 Dense-field representations

For the correspondences and occlusions defined in section 4.2, several dense-field representations C and O have been developed. For segmentation purposes, additional edge-based segmentation fields S and region-based fields R have been proposed. Table 4.1 shows a list of fields used by several authors in their and our notation. We will now have a close look at each of these fields in sections 4.5.1 and 4.5.2. In section 4.5.3 we will discuss some additional fields that are rare or have not been used at all up to now.

		I_A	I_B	C_A	O_A	S_{CA}	R_A
[Konr92]	Konrad & Dubois ’92	g_{t-}	g_{t+}	D_t		L_t	
[Chan94]	Chang et al. ’94	g	g'	u,v			x
[Teka95b]	Tekalp ’95	g_k	g_{k-1}	d	O	l	
[Woo96]	Woo & Ortega ’96	F^l	F^r	D	Φ		
[Stil97]	Stiller ’97	g_t	g_{t+1}	d_t	*see text*		l_t

Table 4.1 Fields for luminance, correspondence, occlusion, discontinuity and segmentation.

4.5.1 Correspondence and occlusion fields

The occlusion points, the real and the pseudo-correspondences can be represented by several pixel-dense fields. They are all defined on the pixel lattice Λ_P (see Figure 4.18). The lattices of the images I_A and I_B are denoted by Λ_{PA} and Λ_{PB}, respectively.

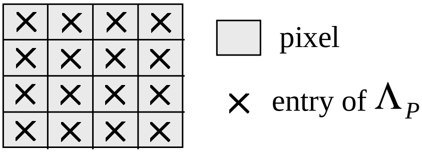


Figure 4.18 The pixel lattice Λ_P.

The correspondence fields C that are mostly used are defined on one of the images lattices Λ_{PA}, Λ_{PB} [Chan94, Rede98d, Stil97, Teka95b, Woo96, Zhan93]. The C_A field is depicted in Figure 4.19. Each entry $C_A(P_A)$ contains a vector with its starting point at the entry P_A on the lattice Λ_{PA}. For pixel-accurate correspondences, the endpoint of the vector lies on the lattice Λ_{PB}. For subpixel accuracy, the vectors end on the continuous domain Λ^*_{PB}.

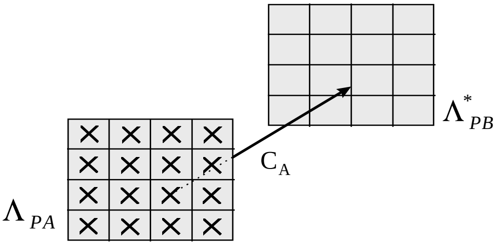


Figure 4.19 The C_A correspondence field.

Most applications benefit from subpixel accuracy, which is reflected in the number of subpixel estimation algorithms that have been developed [Chan94, Stil97, Teka95b, Woo96, Zhan93]. For subpixel accuracy, the luminance of the images has to be interpolated to the continuous domain Λ^*_P. In [Konr92] it is found experimentally that the specific choice of the interpolation filter does not have much influence on the estimated correspondences.

If (x_A,y_A) and (x_B,y_B) are the coordinates of a corresponding pixel pair, the value of the correspondence field C_A is:

$$C_A(x_A,y_A)=\begin{bmatrix} x_B-x_A \\ y_B-y_A \end{bmatrix} \tag{4.4}$$

The value represents the 2-D (vector) displacement of the projection of a scene point between image A and image B. Depending on whether the estimation is performed with pixel or subpixel accuracy, the components of C are integer or real valued. For parallel image pairs, the y component will always be zero and thus does not have to be included. The correspondence vector fields C_A and C_B then reduce to the scalar disparity fields D_A and D_B.

The C_A field can both represent real correspondences between P_A and P_B and pseudo correspondences from P_A to Q_B. In the latter case P_A is an occlusion point. The presence of occlusion points can be represented by the occlusion field O_A:

$$O_A(P_A)=\begin{cases} 0 & \begin{array}{l} P_A \text{ is visible in image B} \\ C_A(P_A) \text{ is a real correspondence} \end{array} \\ 1 & \begin{array}{l} P_A \text{ is an occlusion point} \\ C_A(P_A) \text{ is a pseudo-correspondence} \end{array} \end{cases} \tag{4.5}$$

Figure 4.20 shows the binary occlusion fields O_A and O_B.

When no occlusions are taken into account [Konr92], the C_A field suffices in the modeling process since it is able to represent all real correspondences. If occlusions are taken into account but no pseudo-correspondences are estimated, the C_A field contains all real correspondences and a number of undefined vectors [Teka95b]. Only in [Stil97], all real and pseudo-correspondences from A to B, are estimated, making full use of C_A.

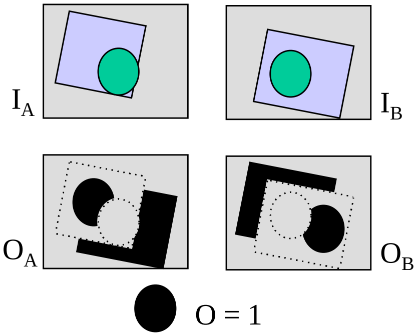


Figure 4.20 The occlusion fields O_A and O_B.

The introduction of both C_A and C_B fields simultaneously enables the estimation of all pseudo-correspondences. This is useful in several applications as discussed in the definition of pseudo-correspondence. Previously, both fields have been estimated separately to remove outliers in real correspondences [Ohm98, Pano98a]. At this point, no attempt has been made yet to estimate both simultaneously.

For parallel image pairs, all real correspondences and both occlusion fields O_A and O_B can be represented by one field, the so-called chain map [Rede97d]. The chain map is applicable if pixel accuracy is used and an additional ordering constraint, discussed in section 4.6.6, holds [Cox96, Fran96, Rede98a, Tsen95]. The chain map itself will be discussed more thoroughly in Chapter 6.

The C_M field used in [Rede97b, Rede98a, Rede99c] is similar to the C_A field, but it is defined on a different domain: Λ_{PM}. It is the pixel grid of a virtual image centered between image A and image B (see Figure 4.21). In [Konr92] a more general case is considered where M is placed at an arbitrary position in between A and B.

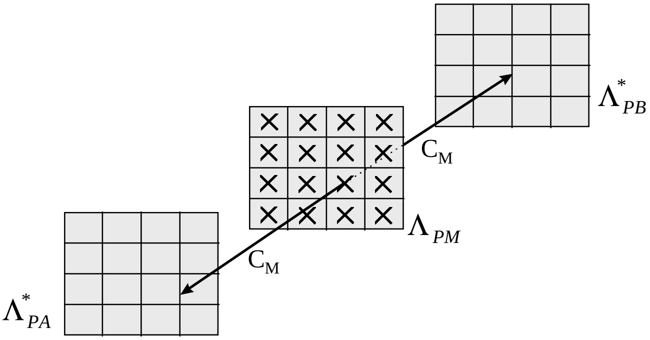


Figure 4.21 The C_M correspondence field.

The value of the C_M field is defined similar to (4.4):

$$C_M(x_M, y_M) = \begin{bmatrix} \dfrac{x_B - x_A}{2} \\ \dfrac{y_B - y_A}{2} \end{bmatrix} \tag{4.6}$$

with

$$\begin{bmatrix} x_M \\ y_M \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_A + x_B \\ y_A + y_B \end{bmatrix} \tag{4.7}$$

The C_M field is attractive because of its symmetry in the A and B images. In parallel image pairs, we have:

$$y_M = y_A = y_B \tag{4.8}$$

and the y component of the field vanishes. Then C_M reduces to the scalar field D_M:

$$D_M(x_M, y_M) = \tfrac{1}{2}(x_B - x_A) \tag{4.9}$$

According to (2.37) in section 2.4.9, we also find $D_M > 0$. This field is particularly interesting due to the simplicity of the triangulation procedure given by (2.49):

$$\begin{bmatrix} P^x \\ P^y \\ P^z \end{bmatrix} = \frac{\tfrac{1}{2} b}{D_M(x_M, y_M) s_x} \begin{bmatrix} (x_M - \tfrac{1}{2} N_x) s_x \\ -(y_M - \tfrac{1}{2} N_y) s_y \\ -f \end{bmatrix} \tag{4.10}$$

Here b, f and s_x, s_y are the parallel camera parameters as outlined in section 2.4.9 (camera baseline, focal length and pixel size respectively), while N_x, N_y is the image size.

The C_M and D_M fields do not allow for an easy incorporation of asymmetric phenomena, such as occlusions and pseudo-correspondences. The field can be used in applications where these phenomena do not play an important role, for example in 3-D face model acquisition from stereo images [Rede99c] and for 3-D videoconferencing (Chapter 6).

In some cases the C_M field cannot represent the real correspondences. A worst-case example is when image B is an 180° rotated version of image A. Then all vectors intersect in the center of Λ_{PM} and C_M can only retain one of them. For the D_M disparity field for parallel image pairs, such an image rotation cannot occur by definition. However, a similar intersection of two (or more) correspondences on Λ_{PM} is possible. This requires that two correspondences on the same scanline (equal y_M) result in equal x_M via (4.7).

We can be sure to avoid this situation by the so-called ordering constraint. This means that all scene points must appear in the A and B images in the same horizontal order. For two scene points P_1 and P_2 that both project to scan line y_M, this means that the interdistance of their projections in the A image, Δx_A, has the same sign as their interdistance in the right image Δx_B. If this is applied to (4.7) and (4.9) we obtain:

$$\left|\frac{\partial D_M}{\partial x_M}\right| \leq 1 \tag{4.11}$$

Often the ordering constraint is already incorporated in the disparity estimation algorithm, since it enables efficient search algorithms, see section 4.8. The ordering constraint is further examined in section 4.6.6.

The D_M field will be used in section 4.9 and in Chapter 6.

4.5.2 Segmentation and correspondence discontinuity fields

Some approaches in correspondence estimation do not model discontinuities [Horn86, Konr92 (MEC algorithm)]. Then, high-quality correspondence estimates are possible if the scene does not contain more than one object of interest, such as face acquisition from stereo [Rede99c].

Segmentation and correspondence discontinuities need to be introduced for image pairs with multiple objects. This has led to the introduction of correspondence discontinuity fields S_C, often called line fields [Konr92, Teka95b, Zhan93], and object label fields R [Chan94, Stil97].

Figure 4.22 illustrates the discontinuities S_C in the correspondence fields C in case of a simple scene with two objects in front of a background.

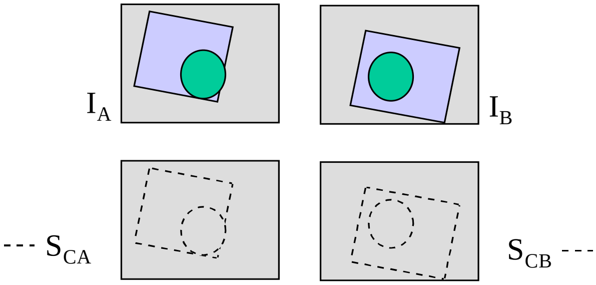


Figure 4.22 Discontinuities in correspondence fields.

Obviously, the discontinuities coincide with the object boundaries. As Figure 4.20 depicts, object boundaries often coincide with boundaries of occlusion areas as well. In [Heit93] experimental results indicate that the incorporation of S or R fields is useful only if occlusion fields O are also taken into account.

The discontinuity fields S are edge based, for which a domain has to be defined. A widely used domain is defined as all sites between two pixels that are four-connected neighbors, denoted by Λ_{S4}, shown in Figure 4.23.

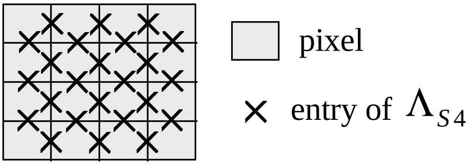


Figure 4.23 Four-connected edge domain Λ_{S4}.

Clearly, Λ_{S4} contains two different kinds of sites, corresponding to horizontal edges between upper and lower pixels and vertical edges between left and right pixels.

The discontinuity fields S normally have binary values. A “0” indicates continuity, a “1” a discontinuity, see Figure 4.24.

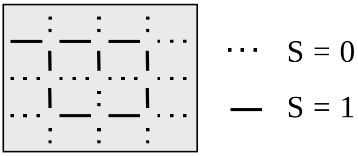
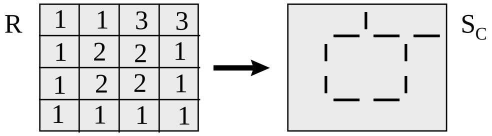


Figure 4.24 An edge-based correspondence discontinuity field S.

Region-based segmentation fields R contain labels for each pixel in the image lattice Λ_P. In [Stil97] a label field R_A is introduced containing natural numbers. Each region of pixels sharing the same label represents a region that is smooth both in the luminance and correspondence fields. In [Chan94, Stil97] a correspondence discontinuity field S_{CA} is derived from a label field R_A as depicted in Figure 4.25. In [Chan94] the R_A field is only used for this purpose, while in [Stil97] additional ordering information is included which allows for the analytic derivation of the occlusion field O_A, using the C_A field as well.



1	1	3	3
1	2	2	1
1	2	2	1
1	1	1	1

Figure 4.25 Given a label field R we can extract a correspondence discontinuity field S_C from it by simply comparing neighboring labels.

A major difference between the S and R fields is that R fields cannot model the open curves as shown within the square in Figure 4.24. These open curves may appear in real images, however, as is shown in Figure 4.26. A single object consisting of a pyramid attached to a plane is recorded by a stereo camera. The fact that the object partly occludes itself in image A leads to open curves of correspondence discontinuities in A.

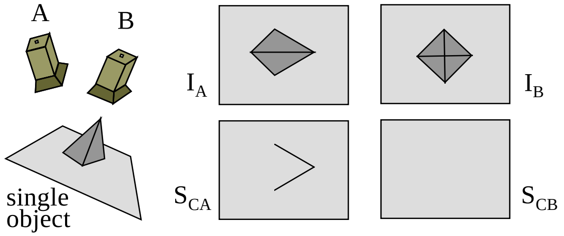


Figure 4.26 Open curves of correspondence discontinuities.

4.5.3 Special fields

Next we will discuss several special fields that have appeared in CE algorithms. In most cases, the algorithms do not (yet) estimate the field pixel-dense and simultaneously.

Image noise

In [Brai95b], estimation of correspondences is combined with image restoration. Besides a correspondence field, an image noise field is included. The estimation is performed recursively by scanning the image.

Specular scene reflectivity

In [Pede95] a ‘field’ is introduced for specular reflectively of the scene in order to account for large luminance differences in the image pair. The approach is feature-based, which yields sparse results.

Correspondence fields for multiple images

In this chapter we deal only with pairs of images. Image sequences, both in temporal and spatial (multiple camera) directions, can however be used to apply additional consistency constraints. This requires the simultaneous estimation of as many dense correspondence fields as there are images in the sequence. This yields a tremendous computational load. For this reason probably, no actual attempt has been made in this direction.

In [Matt89] correspondences are estimated in a sequence, where pairs of images $(t, t+\Delta t)$ are treated one by one. The results are integrated (enhanced) by a Kalman filter. Recursive approaches apply temporal consistency constraints [Stil97] to enhance the estimation in the current image pair on the basis of the previously estimated fields. In [Patr97, Tsai97] the recursive approach is applied on combined temporal/spatial image quadruples. In [Pede97b] spatial image triples are used to obtain accurate feature-based correspondences from edges of curved objects. In [Gram98] multi-camera spatial images are used. Due to a specific camera setup (all in one line), however, a single correspondence field is sufficient in the estimation process.

Epipolar geometry with uncalibrated image pairs

As discussed in section 4.2, the strong epipolar constraint can be applied in a spatial image pair if the cameras are calibrated. For a spatial pair from uncalibrated cameras, the epipolar constraint can still be imposed if the geometry is estimated along with the correspondences. In [Poll98] the pinhole camera geometry is estimated in a preprocessing step using sparse feature (corner) detection and matching. In the simultaneous estimation of dense-field correspondences in uncalibrated spatial image pairs, the epipolar constraint has been applied recently [Rede98d] for cameras with lens distortion. In this approach a field is estimated simultaneously that models the angle of the local tangent to the epipolar lines (see Figure 4.27).

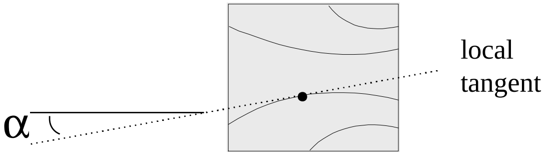


Figure 4.27 Epipolar geometry field.

The curvature of the epipolar lines is interpreted as lens distortion. It is extracted from both images and then penalized. The advantage of this approach is that it does not require feature extraction and estimation of predefined distortion parameters in a preprocessing step. To apply the epipolar constraint in temporal pairs with multiple objects, one needs fields both for epipolar geometry E and its discontinuities S_E, visible in Figure 4.12. Further, a region based segmentation field R_E may be used to group isolated parts in the image into one rigid object, e.g. the rod in Figure 4.12. A similar method has been used in object rigidity checking on the basis of a sparse set of correspondences [Reyn96].

4.6 Joint probability model for the fields

The design of a joint probability model for several dense fields is by no means an easy task. In general, the modeling process is decomposed at two levels. First, by applying the Bayes rule, the fields can be modeled one at a time. Secondly, one can obtain the global model of each of these fields by combining many equal, simple local models. These models assume independence of all entries in a field or dependence only in a small neighborhood reflecting the Markov property, see Appendix C.

As an example we take the approach of [Teka95b], in which the following joint probability is modeled:

$$p_{C_A,S_{CA},O_A|I_A,I_B} \tag{4.12}$$

With the Bayes rule, the joint model is decomposed in several single field models:

$$\frac{p_{I_A|C_A,O_A,I_B} \cdot p_{C_A|S_{CA}} \cdot p_{O_A} \cdot p_{S_{CA}|I_B}}{p_{I_A|I_B}} \tag{4.13}$$

In this decomposition several independencies among the fields are assumed. Table 4.2 shows the joint probability models and Bayes decompositions for the fields in Table 4.1.

Author	Joint model	Bayes decomposition
[Konr92] Konrad & Dubois ’92	$p_{C_A,S_{CA}\|I_A,I_B}$	$p^{-1}_{I_B\|I_A}\ p_{I_B\|C_A,S_{CA},I_A}\ p_{C_A\|S_{CA}}\ p_{S_{CA}\|I_A}$
[Chan94] Chang et al. ’94	$p_{C_A,R_A\|I_A,I_B}$	$p^{-1}_{I_B\|I_A}\ p_{I_B\|C_A,R_A,I_A}\ p_{C_A\|R_A}\ p_{R_A}$
[Teka95b] Tekalp ’95	$p_{C_A,S_{CA},O_A\|I_A,I_B}$	$p^{-1}_{I_A\|I_B}\ p_{I_A\|C_A,O_A,I_B}\ p_{C_A\|S_{CA}}\ p_{O_A}\ p_{S_{CA}\|I_B}$
[Woo96] Woo & Ortega ’96	$p_{C_A,O_A\|I_A,I_B}$	$p^{-1}_{I_A\|I_B}\ p_{I_A\|C_A,O_A,I_B}\ p_{C_A\|O_A}\ p_{O_A}$
[Stil97] Stiller ’97	$p_{C_A,R_A\|I_A,I_B}$	$p^{-1}_{I_B\|I_A}\ p_{I_B\|C_A,R_A,I_A}\ p_{C_A,R_A\|I_A}$

Table 4.2 Joint probability models and Bayes decompositions.

Each of the Bayes factors represents a specific photometric or geometric model, or a combination of them. Examples of photometric models are the Constant Image Brightness (CIB) assumption and its deviations. Geometric models involve continuity and smoothness of the correspondence field.

We will now present several commonly used models and then combine them into a joint model. Finally we discuss some helpful but specialized models that are not yet used in the Bayesian methods, or are used either rarely or in some limited form.

4.6.1 Image luminance models

The factor in the denominator of (4.12) is a constant given that we have observed the images i_A and i_B. In steps 3 and 4 in the framework, criteria for best solutions and search algorithms are selected that do not need the actual value. Thus this factor is never modeled [Chan94, Konr92, Stil97, Teka95b, Woo96].

The first factor in the numerator in (4.12) is similar to the second factor in all Bayes decompositions in Table 4.2. It has the form:

$$p_{I_A|I_B,C\ldots\ldots} \tag{4.14}$$

This factor represents the probability of the luminance of the A image, given the I_B image, the correspondence and other fields.

We will now discuss three ingredients of the luminance models and then combine these.

Pixel independence

All current models for (4.14) assume that luminance is a field with independent entries:

$$p_{I_A|.....} = \prod_{\substack{\text{all pixels}\\ \text{in image } A}} p_{\text{pixel in } A|.....} \tag{4.15}$$

Constant Image Brightness (CIB) and deviations

The basic tool for (4.15) is the Constant Image Brightness assumption, which states that each pair of corresponding pixels P_A and P_B have equal luminance:

$$I_A(P_A) = I_B(P_B) \tag{4.16}$$

The CIB assumption is valid if the cameras are noiseless and all objects have diffuse reflection properties. Additionally, in a spatial image pair the cameras should be photometrically equal. In a temporal pair the photometry of the camera is not allowed to change over time and light sources are not allowed to move with respect to other objects.

All current correspondence estimation algorithms assume CIB as a starting point and model the deviations to some extent. Mostly, the causes for CIB deviations are modeled together by a zero-mean Gaussian [Chan94, Konr92, Teka95b, Woo96]:

$$p(\Delta I) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\Delta I^2}{2\sigma^2}} \tag{4.17}$$

In [Stil97] a generalized Gaussian is used, with shape and variance parameters estimated from the images. The shape parameters obtained suggest that a Laplacian outperforms a Gaussian, a result which has been found earlier in [Mara89]. In [Pede95] deviations due to specular reflection of scene surfaces are modeled in a feature-based approach for correspondence estimation. Photometric differences in cameras can be accounted for in advance by means of luminance histogram warping [Cox95].

Occlusions

For occlusion points in image A, no relation such as (4.16) or (4.17) can be established. In [Stil97] the luminance is then modeled with a uniform probability distribution over all grey levels:

$$p(I) = \frac{1}{N_{greylevel}} \tag{4.18}$$

The complete model

We will derive a complete luminance model for the image I_A given the image I_B, the correspondence field C_A and the occlusion field O_A, see Figure 4.28. According to (4.14) we model each pixel independently:

$$p_{I_A|C_A,O_A,I_B} = \prod_{P_A \in \Lambda_{PA}} p_{I_A(P_A)|C_A,O_A,I_B} \tag{4.19}$$

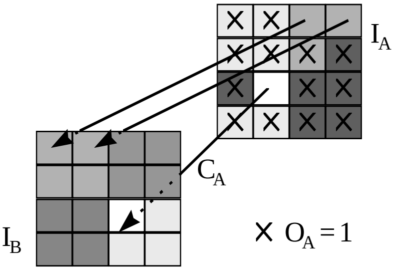


Figure 4.28 Predicting the luminance of image I_A from image I_B using correspondence field C_A and occlusion field O_A.

To incorporate the occlusion pixel model (4.18) in (4.19), we need to know which pixels in A are occlusion points, and for the Gaussian CIB deviation model (4.17) which pixels are not occlusion points. This information is contained in occlusion field O_A (4.5). In (4.17) the ΔI term refers to the luminance difference of a pair of corresponding pixels in A and B. For each non occlusion pixel in image A, we need a real correspondence vector that originates from that pixel. These vectors are the real correspondence vectors contained in the C_A field. If we apply (4.17) and (4.18) in (4.19) using the C_A and O_A fields (4.4) and (4.5), we obtain

$$p_{I_A|C_A,O_A,I_B} = \prod_{\substack{P_A \in \Lambda_{PA} \\ \text{with } O_A=1}} \frac{1}{N_{greylevel}} \cdot \prod_{\substack{P_A \in \Lambda_{PA} \\ \text{with } O_A=0}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(I_A(P_A)-I_B(P_A+C_A(P_A))\right)^2}{2\sigma^2}} \tag{4.20}$$

A similar expression is found in [Stil97], on the basis of a generalized Gaussian. In [Konr92], no occlusions are taken into account and only the product series with $O_A = 0$ in (4.20) is obtained. The same holds for [Teka95b] in which the occlusion point model is discarded.

4.6.2 Correspondence smoothness models

Within continuous areas of a correspondence field, it is assumed that the field is also a smooth function of position. Smoothness of correspondence reflects smoothness of scene surfaces.

The most basic smoothness constraint penalizes large values of the spatial derivatives of the correspondence field. This means that differences of neighboring entries in the field are penalized. Gibbs and Markov Random Field models are the tool to take into account such interactions between neighboring field entries, see Appendix C.

An example of a Gibbs Random Field (GRF) model that enforces global smoothness on correspondence field C_A is:

$$U_{C_A} = \alpha \sum_{Q \in \Lambda_{S4A}} \left| C_A\left(P_{Q1}\right) - C_A\left(P_{Q2}\right) \right|^2 \tag{4.21}$$

The GRF is defined in the energy domain U instead of probability p, via $p = e^{-U}/Z$ (see Appendix C). Figure 4.29 illustrates (4.21). For all neighboring entries (P_{Q1}, P_{Q2}) on the Λ_P domain, the cliques, the difference in the C_A entries are squared and added. In (4.21) the cliques are indexed by the entries Q of domain Λ_{S4}. Large fluctuations in the correspondence field yield high energies, which result in a low probability for that field.

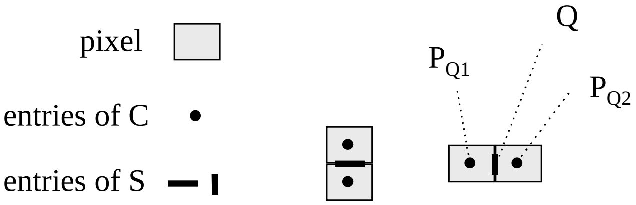


Figure 4.29 Cliques in a GRF for correspondence smoothing.

The scale factor α is used in all approaches to regulate the influence of the smoothness constraint with respect to other constraints. Each constraint has its own scale parameter, which are usually determined experimentally. For simplicity, we will use α for any scale parameter in this section (in the experiments section we will use separate variables).

As shown in Appendix C, the energy U_{CA} in (4.21) results in a Bayes factor:

$$p_{C_A} = \frac{1}{Z} e^{-U_{C_A}} \tag{4.22}$$

As discussed in Appendix C, the value of Z cannot be computed. However, it is a constant and it can be discarded from the modeling process for the same reason as the denominator in (4.12).

In [Li95] the square in (4.21) is replaced by more general functions that more or less incorporate discontinuities without modeling them explicitly.

It is often assumed that smoothness of correspondence C is correlated with smoothness of luminance I. Overviews of these photometric-geometric models can be found in [Enke88, Nage86, Snyd91]. Basically, these models relax smoothness constraints across luminance edges, resulting in so-called oriented smoothness constraints.

Smoothing the correspondence field C_A while taking into account the discontinuities according to S_{CA} can be performed by a compound GRF model involving both fields [Konr92]:

$$U_{C_A|S_{CA}} = \alpha \sum_{Q \in \Lambda_{S4A}} \left| C_A\left(P_{Q1}\right) - C_A\left(P_{Q2}\right) \right|^2 \left(1 - S_{CA}(Q)\right) \tag{4.23}$$

Variations to (4.23) can be found in [Teka95b, Zhan93]. In [Chan94] and [Woo96] a segmentation field R_A is used as in Figure 4.25. In [Woo96] the occlusion field O_A is used as approximation to R_A.

4.6.3 Correspondence discontinuity models

For the discontinuity field S_{CA} in (4.23) several models have been proposed, which can be divided in three different types.

Low number of discontinuities
Discontinuities can be penalized independently for each entry in the field:

$$U_{S_{CA}} = \alpha \sum_{Q \in \Lambda_{S4A}} S_{CA}(Q) \tag{4.24}$$

This enforces a low number of discontinuities. In [Chan94] a model similar to (4.24) is used on the basis of label field R_A and the procedure shown in Figure 4.25.

Coincidence of discontinuities in correspondence and luminance
A second model for discontinuities is that they often coincide with luminance discontinuities [Konr92, Zhan93], reflecting a combined photometric-geometric constraint:

$$U_{S_{CA}|I_A} = \alpha \sum_{Q \in \Lambda_{S4A}} \frac{S_{CA}(Q)}{\left| I_A\left(P_{Q1}\right) - I_A\left(P_{Q2}\right)\right|^2} \tag{4.25}$$

In the case of a zero in the denominator no discontinuity is allowed at Q.

In [Stil97] a discontinuity field is defined on Λ_{S8}, which also includes entries for diagonally neighboring pixels (eight-connectedness). The field is extracted from a label field R_A in a similar way as that in Figure 4.25.

Global connectivity of discontinuities to form curves
The third model for discontinuities is that they form globally connected curves. The connectivity is generally modeled by means of a GRF. Figure 4.30 shows an example of the cliques and their energy in [Teka95b].

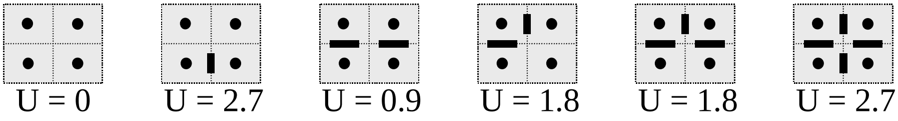


Figure 4.30 Cliques to enforce connectivity of correspondence discontinuities.

In [Konr92] similar cliques are defined, including additional cliques that prohibit single pixels to become surrounded by discontinuities.

4.6.4 Occlusion models

For the occlusion field, two types of models are generally applied.

Low number of occlusions
First the presence of occlusions can be penalized [Teka95a], similar to (4.24) for S_C:

$$U_{O_A} = \alpha \sum_{P_A \in \Lambda_{PA}} O_A\left(P_A\right) \tag{4.26}$$

Penalizing occlusion points promotes the detection of real correspondences in an image pair.

Global connectivity to form areas
The second model of occlusions encourages connectivity of occlusion points [Woo96]. Such a model can be obtained by applying (4.24) on discontinuities S_O extracted from the O field as depicted in Figure 4.25. In this way the occlusion points are forced to form connected areas, as shown in Figure 4.20.

4.6.5 Combination into a joint model

To combine the luminance model (4.20), the correspondence smoothness model (4.23), the correspondence discontinuity model (4.24) and the occlusion model (4.26) into a joint probability model, we will convert the latter three to the probability domain. For the occlusion model, this results in:

$$p_{O_A} = \frac{1}{Z} e^{-U_{O_A}} \tag{4.27}$$

Similar to (4.22), the partition function Z is a constant and can be neglected. For the discontinuity adaptive correspondence smoothness model (4.23) we have:

$$p_{C_A|S_{CA}} = \frac{1}{Z} e^{-U_{C_A|S_A}} \tag{4.28}$$

In this case, the partition function Z is *not* a constant, but a function of the conditioning field S_{CA} [Gema84], which itself is not constant during estimation. This is often neglected, as is done in [Chan94, Konr92,Teka95b, Woo96], even though it is inconsistent with the application of the Bayes rule.

In general, partition functions are not constant when two or more output fields interact with each other and are modeled in the energy domain. A way to circumvent this is not to apply the Bayes rule on those fields. Instead, we combine the energies of (4.23) and (4.24) to form a model that is joint in these two fields:

$$p_{C_A,S_{CA}|I_A} = \frac{1}{Z} e^{-U_{C_A,S_A}+U_{S_A|I_A}} \tag{4.29}$$

In [Stil97] this is applied on the correspondence and segmentation fields C_A and R_A.

At this point, we face a similar challenge for a different reason. The models (4.20) and (4.29) contain circular dependencies of I_A and C_A, and thus these equations cannot be combined to form a joint model by means of the Bayes rule. If an attempt is made, the wrong image B appears in one of the Bayes factors (see Table 4.2). If the C_M field is used, it is not clear which image should appear on which side.

A general solution is to transform also the luminance model to the energy domain via $U = -\ln p$, and then to add all energies to form a joint model:

$$p'_{C_A,S_{CA},O_A,I_A,I_B} = \frac{1}{Z} e^{-\left(U_{I_A,C_A,O_A,I_B}+U_{C_A,S_{CA}}+U_{S_A,I_A}+U_{O_A}\right)} \tag{4.30}$$

For the energy terms in (4.30), no relation with any Bayes factor can be established for the decomposition of *p*’. In the optical-flow-based approach in [Heit93] a joint model is designed similarly.

In (4.30) the joint model is constructed by adding energies freely instead of by using the Bayes rule. This is at the cost of some explicitness in the modeling process, but inevitable since it allows for the integration of several useful models.

4.6.6 Special models

In section 4.5.3 we discussed several special fields. If introduced in CE algorithms, an additional model is needed also. Models for those special fields can be found via the references in section 4.5.3. Here we discuss two special models, that are related to normal fields such as correspondence and discontinuities fields.

Ordering constraint

A strong and useful relation exists between correspondence, its discontinuities and epipolar geometry. If there is an interval without correspondence discontinuities along a pair of conjugate epipolar lines, then the ordering constraint holds at this interval. The constraint means that scene points appear in the same order along the intervals in *A* and *B*, see Figure 4.31.

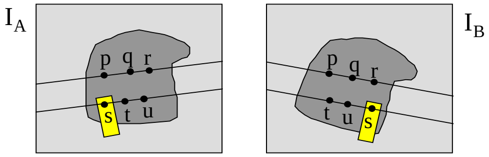


Figure 4.31 The ordering constraint.

The ordering constraint does not always hold across discontinuities. Figure 4.31 shows an example in which a thin object moves fast in front of a background (temporal pair), or is present in front of it and recorded by a stereo camera with large baseline (spatial pair).

In disparity estimation in parallel image pairs, the ordering constraint is often applied globally, regardless of discontinuities [Cox96, Rede98a, Tsen95]. This simplifies the algorithms because a strong constraint can be applied without the need for discontinuity estimation. Additionally, it enables the use of the deterministic search algorithm dynamic programming (DP), see section 4.8. In temporal and uncalibrated spatial image pairs, the ordering constraint has not yet been applied.

Round-about constraint

The round-about constraint holds among several correspondence fields for multiple images of the same scene. Two fields may e.g. be used in a stereo image pair to estimate all pseudo-correspondences (section 4.5.1), multiple fields may be used for CE within multiple images (section 4.5.3). In such cases, a consistency relation between the fields exists for any scene point that is visible in all the images.

Consider the simple case of two images A and B and two correspondence fields C_A and C_B. If we find for scene point P that $C_A(P_A) = P_B$, then we must have $C_B(P_B) = P_A$. The constraint can be most easily described by $C_B(C_A(P_A)) = P_A$. A loop is formed between the A and B images, giving the constraint its name. In the combined temporal/spatial approach of [Patr97], image quadruples $I_A(t)$, $I_B(t)$, $I_A(t,t+\Delta t)$, $I_B(t,t+\Delta t)$ are used. If we call these images I_A, I_B, I_C and I_D, and use four correspondence fields $C_{A\rightarrow B}$, $C_{B\rightarrow C}$, $C_{C\rightarrow D}$, $C_{D\rightarrow A}$, the round-about constraint may look like this. If $C_{A\rightarrow B}(P_A)=P_B$, $C_{B\rightarrow C}(P_B)=P_C$ and $C_{C\rightarrow D}(P_C)=P_D$, then we must have $C_{D\rightarrow A}(P_D)=P_A$.

The constraint can be used to estimate one of the fields given the other fields, but only in terms of the real correspondences (pseudo-correspondences are unique for each of the fields). The specific form of the constraint depends on the way the correspondence fields are arranged.

4.7 Criteria for best solutions

The best solution f_{SOL} can be defined in many ways. In the area of simultaneous estimation of dense correspondence fields, three criteria are used: the Maximum A Posteriori (MAP) criterion, the Maximum Likelihood (ML) criterion and the mean field (MF) criterion.

4.7.1 Maximum a Posteriori criterion (MAP)

The most widely used criterion is the MAP criterion [Chan94, Konr92, Stil97, Teka95b, Woo96]:

$$f_{MAP} = \arg\max_{\hat{f}} p_{F|I_A,I_B}\left(\hat{f}, i_A, i_B\right) \tag{4.31}$$

The MAP criterion selects the solution that has highest probability given the observed images i_A and i_B. Since these are constants in the maximization in (4.31) we have:

$$p_{F|I_A,I_B} \propto p_{F,I_A,I_B} \propto p_{F,I_B|I_A} \propto p_{F,I_A|I_B} \tag{4.32}$$

The MAP solution can be obtained by maximizing any of the probability functions in (4.32).

4.7.2 Maximum likelihood criterion

This criterion is defined as:

$$f_{ML} = \arg\max_{\hat{f}} p_{I_A,I_B|F}\left(i_A, i_B, \hat{f}\right) \tag{4.33}$$

On its own, this formula does not make any sense. The observed images i_A and i_B are used as argument of a function that accepts possible images as arguments, and vice versa for the other argument. Since

$$p_{I_A,I_B|F} = \frac{p_{F,I_A,I_B}}{p_F} \tag{4.34}$$

the ML criterion makes sense as special case of the MAP criterion, see (4.31) and (4.32), provided that the marginal probability function p_F of the fields to be estimated, the prior model for F, is a constant. This excludes any correspondence smoothness model, occlusion model or discontinuity model. Therefore, the ML criterion cannot be used in correspondence estimation. In [Cox96] an ML algorithm is presented that implicitly penalizes occlusions, so it is not a real ML algorithm.

4.7.3 Mean field criterion (MF)

The MF criterion is used less frequently than the MAP criterion and is defined as:

$$f_{MF} = \int_{\hat{f}} \hat{f} \cdot p_{F|I_A,I_B}\left(\hat{f}, i_A, i_B\right) d\hat{f} \tag{4.35}$$

It yields the average or expected solution, which is equal to the first moment of the probability function on the output variables conditioned by the observed image pair. It can only be used for continuous output variables, since discrete variables such as binary occlusion and discontinuity fields cannot be averaged.

This criterion is used in [Zhan93], and in [Konr92], where it is called mean expected cost (MEC) criterion.

4.7.4 Summary

From the above, it is clear that the ML criterion is not useful for our application. The remaining MAP and MF criteria are special cases of a parameterized family of criteria [Ther92]. The MF criterion has been reported to yield results similar to the MAP criterion whenever both criteria can be applied [Konr92].

Whenever discrete fields are used (occlusion or segmentation/discontinuity fields), only the MAP criterion is applicable. If only continuous fields are used, both MAP and MF criteria are available and these lead to similar results. The choice can then be made based on other terms, e.g. the availability of an efficient search algorithm.

4.8 Search algorithms

Here we will discuss several search algorithms for the MAP and MF criteria. After that, we discuss the hierarchical approach that can be used in combination with the search algorithms to speed up computation time and to increase the quality of the estimated fields.

4.8.1 MAP search algorithms

Since the dimensionality of (4.31) is extremely large, the probabilities get extremely small. Even for the actual MAP solution f_{MAP} it may be in the order of $10^{-1,000,000}$. Therefore energy is used rather than probability in the numerical evaluation of (4.31). As an example we will consider the joint model in (4.32):

$$p_{F,I_A,I_B} = \frac{1}{Z} e^{-U_{F,I_A,I_B}} \tag{4.36}$$

Maximization of probability is equivalent to minimization of energy:

$$f_{MAP} = \arg\min_{\hat{f}} U_{F,I_A,I_B}\left(\hat{f}, i_A, i_B\right) \tag{4.37}$$

The partition function Z is a constant and has been removed in (4.37). Many search algorithms are available for the minimization in (4.37). They are either exact or approximate, and either deterministic or stochastic.

The most well-known technique for these kinds of minimizations is the downhill or gradient descent method [Pres92]. It is a deterministic method that easily gets stuck in local minima.

To avoid local minima, stochastic methods are used, such as simulated annealing (SA). In SA, an estimate to the solution is perturbed at random. Better estimates (less energy) are always accepted, worse estimates are accepted now and then, governed by a temperature parameter. If one decreases the temperature from T_0 to zero infinitely slowly [Gema84], the exact solution to (4.37) is reached. In practice, the temperature is lowered much faster and an approximation is obtained.

To use the SA algorithm, one has to define a temperature-cooling schedule and a random perturbation generator. At this point there are no general rules to help the designer. In [Stil97] a cooling schedule is chosen that decreases exponentially. A table is presented with several perturbations, such as small changes in the correspondence fields and flipping of the binary values of the occlusion and discontinuity fields.

Many different versions of SA have been presented, e.g. the Metropolis algorithm [Teka95b], Iterated Conditional Modes (ICM) [Chan94, Heit93, Teka95b], and the so-called Gibbs sampler methods [Gema84]. The interested reader is referred to the specific articles for details.

The only exact and deterministic algorithm for the MAP solution is the Dynamic Programming (DP) or Viterbi algorithm [Cox96, Fran96, Gram98, Inti94, Rede98a, Tsen95]. It can be used to estimate disparity in parallel image pairs. It requires that (4.37) is separable in all scan lines, which excludes interactions between scan lines, such as smoothing. It is especially efficient if the ordering constraint (see sections 4.5.1 and 4.6.6) is applied.

Figure 4.32 shows the MAP solutions for the D_M field, obtained by an exact DP algorithm without vertical smoothing, and by an approximate SA algorithm including vertical smoothing that obtains the dense field simultaneously. Clearly, the result from the SA algorithm shows consistency in the vertical direction, while the DP result appears quite random vertically. This effect is strongest in areas without texture.

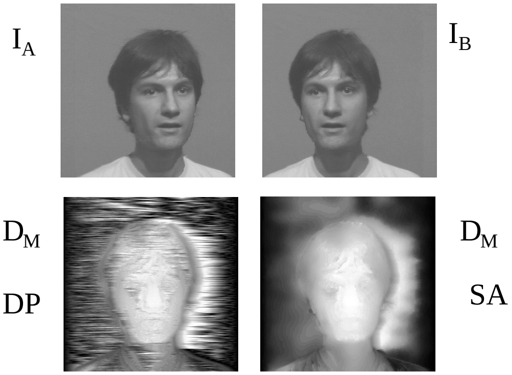


Figure 4.32 MAP solutions obtained by DP and SA search algorithms.

Adaptations to the DP algorithm have been made in [Ohta85, Rede98b], which include vertical consistency to some extent, without the need for simultaneous estimation. Recursive estimators have been derived that have similar properties [Brai95b]. These approaches have non-separable (4.37) as a starting point, that is, a general MRF model. In all these algorithms that combine vertical consistency with causal search algorithms, the modeling and the search algorithm are intertwined. Therefore it is less clear which model is effectively used.

Genetic algorithms (GA) have been used for correspondence estimation. In [Fran95] the estimation is done separately for each scan line. For dense simultaneous estimation the GA approach is not feasible since it requires several solution estimates to be maintained at the same time. This demands a tremendous amount of memory and computational power.

4.8.2 MF search algorithms

The Mean Field Theory (MFT) is used in [Zhan93] to obtain the MF solution. It is based on the following approximation to (4.35):

$$f_{single} \approx \int_{\hat{f}_{single}} \hat{f}_{single} \cdot p_{F_{single}|F_{rest},I_A,I_B}\left(\hat{f}_{single}, f_{rest}, i_A, i_B\right) d\hat{f}_{single} \qquad (4.38)$$

It means that if the mean solution of all fields f_{rest} is given except for a single entry of one field f_{single}, we can obtain an approximation to this single variable. Evaluation of (4.38) only requires integration over a single variable of the output space, while (4.35) requires integration over the entire solution space. The marginal probability model in (4.38) can easily be obtained from joint models on the basis of Gibbs Markov random fields, see Appendix C.

Given an approximation of the complete solution, we can obtain a better approximation of each single variable with (4.38), in order to obtain the next approximation of the complete solution.

In [Konr92] a different technique is used to obtain the MF solution, which is based on the so-called Gibbs sampler [Gema84]. A Gibbs sampler provides a sequence of different realizations $f_{Gibbs,i}$ of the fields to be estimated, according to the probability model $p_{F|I_A,I_B}$ in (4.35). A statistical average of N of these realizations is an approximation to the mean solution:

$$f_{MF} \approx \frac{1}{N}\sum_{i=1}^{N} f_{Gibbs,i} \qquad (4.39)$$

4.8.3 Hierarchical approach

The MAP and MF search algorithms for dense fields yield a large computational burden. Although stochastic methods are designed to avoid local minima, the restrictions for a feasible implementation (fast cooling schedules and a low number of iterations) still lead to problems with local minima.

A general approach that provides faster convergence and at the same time avoids local minima is the hierarchical approach. Due to its good results for natural images it is used in a wide variety of correspondence estimation algorithms [Acca95, Chan94, Enke88, Heit93, Konr92, Liu93, Ohm98, Patr97, Rede99c, Stil97].

Figure 4.33 depicts the hierarchical approach. The observed images are downsampled to lower-resolution versions. The original images are at level 0; the resolution decreases with increasing level number. At the lowest resolution level *L* the estimation starts. After estimation, the fields are upsampled to the resolution of level *L*-1. These fields are then used as an initial estimate for the estimation at this level. This continues until estimation is performed at full resolution level 0.

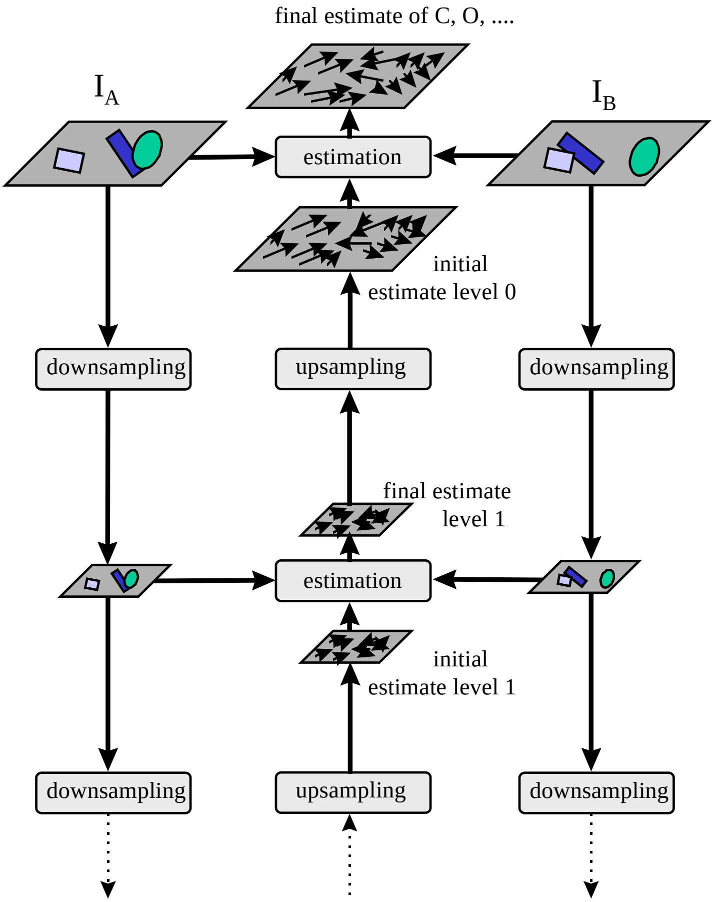


Figure 4.33 Hierarchical estimation.

Many different upsampling, downsampling and estimation schemes can be chosen. This involves the selection of new lattices *Λ* for the lower-resolution fields, suitable filters and possibly level-dependent search algorithms. In most cases, the influence of these choices is small compared to other choices made in the four steps in the Bayesian framework.

The most popular schemes for the lower-resolution lattices are the 2:1 schemes [Konr92, Stil97], in which both x and y axes are subsampled with a factor two. Schemes with non-integer ratios also exist. In [Lund98], the effect of these schemes on computational efficiency is investigated. Many different filters are used for downsampling the images, such as Gaussian filters [Liu93, Patr97] and low-pass FIR filters [Konr92]. In [Stil97] bilinear filters are used to upsample the correspondence fields, and nearest-neighbor interpolation filters to upsample the discrete label fields.

Generally, for the estimation at different levels, the same algorithm is applied at each level. However, some authors include level dependencies, such as increased smoothness constraints [Patr97] or removal of discontinuity fields at lower resolution levels [Konr92]. In [Luet93] special types of Markov Random Field probability models are investigated for which the efficiency of level-independent estimation schemes is optimal. Using other models, however, may still result in near optimal schemes [Stil97].

In literature, we found no reference to the combination of the hierarchical approach and the temperature cooling schedule in simulated annealing approaches. Both regulate the scale or resolution at which perturbations influence the final estimate of the solution. If combined, the SA temperature schedule might not be needed to provide convergence, which simplifies the algorithm substantially.

4.9 Bayesian design of new estimators

In this section we will propose two correspondence estimation algorithms. They are designed to meet the requirements of our 3-D videoconferencing application, which are high (pixel-dense) resolution, sub-pixel accuracy if possible, and reasonable computational load enabling real-time algorithms at this moment or in the near future. In the design we will follow the Bayesian approach using the four steps. We will make full use of our findings in sections 4.4 to 4.8 and show that the four steps enable us to design the estimator in a clear and fast way. Especially, we will use and test the conjecture made in section 4.8.3 about the efficient combination of SA search algorithms and the hierarchical framework.

First we discuss a basic algorithm in section 4.9.1 that uses the parallel camera setup. Experiments are performed in section 4.9.2. Then in section 4.9.3 we propose a modified estimator that can deal with uncalibrated cameras in any camera setup, followed by experiments in section 4.9.4.

4.9.1 Disparity estimator for parallel camera setup

Due to the parallel camera setup, correspondence estimation is reduced to disparity estimation, see section 4.2.3. Figure 4.9.1 shows the first stereo frame of a sequence taken by Heinrich Hertz Institute Berlin. The source material is typical for videoconferencing: a human head plus shoulders in front of a uniform background. The camera baseline was as large as 30 cm to enable high-accuracy 3-D scene acquisition, which yields also large differences between the left and right images.

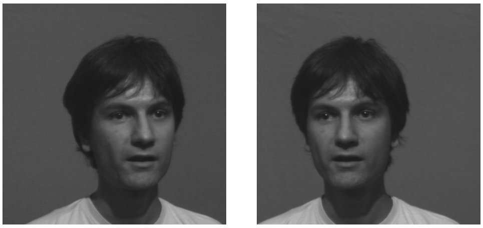


Figure 4.34 Typical stereo images for a 3-D teleconferencing application.

The design of the estimator is divided in the four steps of the Bayesian framework.

Step 1: Definition of fields

We will use three pixel-dense fields: the original images I_A (left) and I_B (right), and the disparity field D_M. In our application, only one object of interest is present, the human subject, and thus we do not introduce segmentation fields. Similarly, as occlusions arise at object edges, we do not include occlusion fields. This will lead to falsely found real correspondences at the ears in Figure 4.9.1, where in principle only pseudo-correspondences can be estimated.

To allow for sub-pixel correspondence estimates, the original images will be interpolated (online) to continuous images by bilinear interpolation, see section 4.5.1. We will use the D_M disparity field with a real-valued x component (it has no y component, see section 4.5.1). We will use D_M and not D_A or D_B to make the aforementioned effect around the ears symmetrical and less noticeable. In section 6.3.4 this effect will be studied in more detail.

Step 2: Joint probability model of all fields

As discussed in section 4.6.5, especially when the D_M (or C_M) field is used, we can only form a joint model by designing all submodels in the energy domain and by adding them freely to form the joint model.

We will use two submodels: one for the luminance of images I_A and I_B, and one for the smoothness of correspondence field C_M. For the luminance model, we will use (4.20) with the D_M field, without occlusions. Further we take the Laplacian instead of the Gaussian, as this was reported to yield better results (see section 4.6.1):

$$U_{\text{luminance}} = \sum_{P_M \in \Lambda_{PM}} \left| I_B\left(P_M + D_M\left(P_M\right)\right) - I_A\left(P_M - D_M\left(P_M\right)\right) \right| \tag{4.40}$$

For the smoothness model we use (4.21) applied to the D_M field:

$$U_{\text{smoothness}} = \sum_{Q \in \Lambda_{S4M}} \left| D_M\left(P_{Q1}\right) - D_M\left(P_{Q2}\right) \right|^2 \tag{4.41}$$

Figure 4.29 shows the meaning of Q and P_{Q1} and P_{Q2}. Effectively, this model penalizes non-zero spatial derivatives of the correspondence field (extracted with [-1 1] and $[-1\ 1]^T$ filters).

Although we do not incorporate an occlusion field for our application, we do include a rudimentary model for occlusions. For very large values of an entry of D_M, a correspondence vector will point outside both the *A* and *B* images, which can be interpreted as a scene point that is visible in virtual image *M* but is occluded in both *A* and *B* images (since it falls out of the images). We introduce a bias for all correspondence vectors to point inside the images:

$$U_{\text{occlusion}} = \sum_{\substack{P_M \in \Lambda_{PM} \\ \text{with } D_M \\ \text{pointing} \\ \text{outside} \\ \text{images}}} 1 \tag{4.42}$$

The joint model is then formed by adding the energies, including weight factors:

$$U_{\text{joint}} = U_{\text{luminance}} + \alpha U_{\text{smoothness}} + \beta U_{\text{occlusion}} \tag{4.43}$$

Experimentally, we have to find the weight coefficients α and β. The conversion to probability via $p = e^{-U}/Z$ will not be necessary, since the next steps will work on energy *U*.

Step 3: The criterion for the best solution

We will use the MAP criterion, since we will use simulated annealing as our search algorithm, see sections 4.7.4 and 4.8.1.

Step 4: The search algorithm

The MAP criterion requires the minimization of (4.43) with respect to the D_M field. For this we will use a simulated annealing (SA) algorithm in combination with the hierarchical approach, see sections 4.8.1 and 4.8.3. We conjectured in section 4.8.3 that the hierarchical approach may take over the effect of the SA temperature cooling schedule. Therefore we select the temperature *T* to be constant.

The basis of our SA algorithm is visiting each entry of the D_M field, to perturb it with a Gaussian random value with some σ, and to calculate the ΔU_{joint} in (4.43) it yields. If ΔU_{joint} is negative, a better solution is found and the perturbation will be accepted permanently. If ΔU_{joint} is positive, a worse solution is found and the perturbation will be accepted only with probability $p = e^{-\Delta U/T}$, as explained in section 4.8.1. The calculation of ΔU_{joint} can in principle be done by calculating U_{joint} in (4.43) completely both before and after the perturbation. This would require the full summation in (4.40), (4.41) and (4.42). In any Markov Random Field model, a perturbation has only local effects on these summations, that is, a single entry of (4.40), four entries of (4.41) and one entry of (4.42). Therefore ΔU_{joint} can be calculated very efficiently. In our implementation we choose to scan all of the field entries with perturbations first, and after that update the whole field simultaneously with all accepted perturbations. In this way, no scan pattern needs to be defined and no field entries have a special role (start or end points in the scanning).

To enable sub-pixel accuracy, we extend the hierarchical approach beyond the pixel-resolution level 0 as indicated in Figure 4.33. After this level we do not increase the resolution, but decrease the σ of the perturbations of the SA algorithm. Experimentally, we have to find values for the temperature T, the sigma σ of the perturbation generator and the number of iterations on each level.

4.9.2 Experiments

For the stereo image pair in Figure 4.34, we have performed a large number of experiments. These lead to the experimentally found values of $\alpha = 3$, $\beta = 100$ for the joint model of step 2, and $\sigma = 2$ and $T = 1$ for the SA algorithm in step 4. The results were quite insensitive to the value of β in the range of 10-1000 and T between 0.1 and 10. The values of α and σ, however, could be changed only about ±0.5 without effecting the results. For larger α or smaller σ the results were too smooth, while for smaller α or larger σ the algorithm did not converge well.

The number of iterations performed on each resolution level in the hierarchical approach was chosen as shown in Table 4.3. For all levels, the number of iterations can be increased without affecting the results, but when the number is decreased, the quality drops. Increasing the number of iterations on the high-resolution levels increases the computational load of the algorithm severely. At the top of the hierarchy more levels can be included, which in theory allows reaching any arbitrary sub-pixel accuracy. However, we assumed that more than 1/8 pixel was not relevant.

Hierarchy level	Resolution	σ	Number of iterations
-4	386x386	0.125	3
-3	386x386	0.25	3
-2	386x386	0.5	3
-1	386x386	1	3
0	386x386	2	10
1	193x193	2	10
2	96x96	2	30
3	48x48	2	30
4	24x24	2	100
5	12x12	2	100
6	6x6	2	100
7	3x3	2	100

Table 4.3 Perturbation noise and number of iterations performed on each resolution level in the hierarchical approach.

Figure 4.35 shows the D_M field on several levels in the hierarchy. The silhouette of the person's head is clearly visible. Subjectively, the indicated disparity or object depth values are in agreement with those to be expected. The values in the background are not correct, but since there is no texture it is not possible to deduce its disparity or depth.

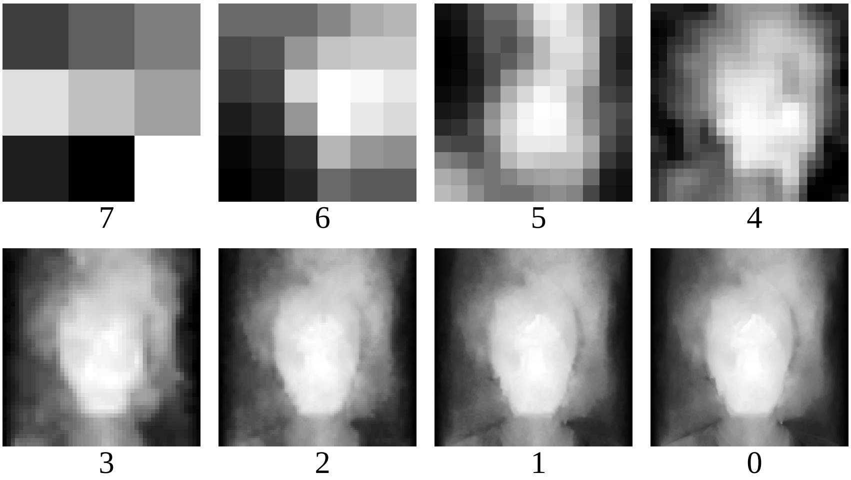


Figure 4.35 The estimated D_M field during several levels in the hierarchy. The final estimate with full resolution is at level 0. Bright areas indicate large disparity (near objects) and dark areas small disparity (distant objects).

We evaluated the quality of the D_M field via interpolated views and 3-D models. Figure 4.36 shows interpolated views using the estimated D_M fields for several frames in the sequence. Figure 4.34 shows the original images for the first frame. The interpolation was done by copying image luminance from the original left and right images along the vectors in the D_M field onto the Λ_{PM} grid, see Figure 4.21. In Chapter 6 the interpolation algorithm is described in detail. Clearly, the interpolated views appear as very natural camera images.

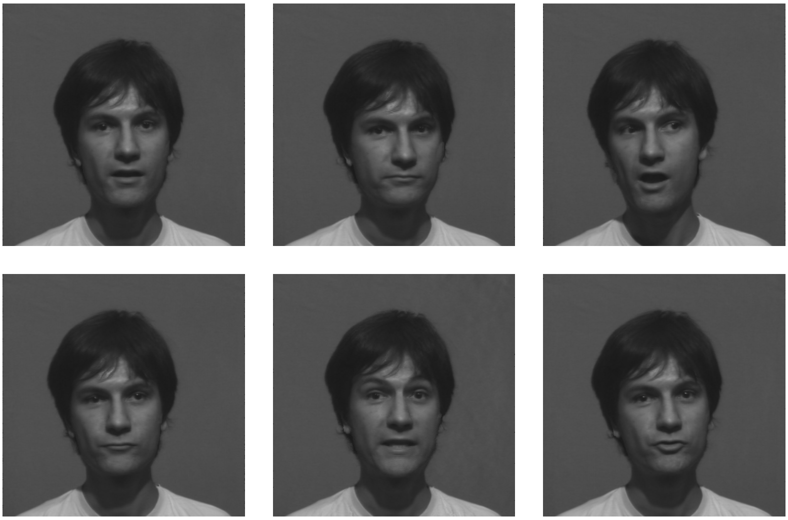


Figure 4.36 Interpolated views using the estimated D_M field for several frames of the original image sequence. Figure 4.34 shows the original images for the first frame.

An important observation is that the simulated annealing algorithm works very well with the constant temperature schedule. As conjectured in section 4.8.3, the hierarchical approach takes over the role of the cooling schedule. The computation time was about 20 seconds per frame on a Silicon Graphics Octane computer. This is orders of magnitude faster than the computation times reported in other Markov Random Field and Simulated Annealing based algorithms [Stil97]. This is ascribed to the only significant difference with these approaches: the new combination of the SA algorithm and the hierarchical approach.

About 5% of the runs of the correspondence estimation algorithm did not converge to the right solution. Such non-convergence percentages are seldomly reported, hence we cannot compare this result with other approaches. Figure 4.37 shows interpolated views for two of the non-converged cases. We could not find values for α and σ that always ensured good convergence. By tuning T (> 1) and the number of iterations in the hierarchy as well, we could always find frame-specific settings for which good results were ensured.

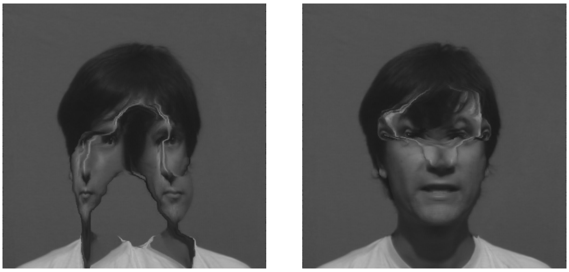


Figure 4.37 Interpolated views for runs of the correspondence algorithm that did not converge to the right solution.

This sequence was accompanied by calibration parameters, and thus we could perform scene triangulation directly. Figure 4.38 show several views of the 3-D model acquired. The ears do not appear well in the 3-D model, but this is to be expected since they cannot be seen in stereo in the original images. Also the background appears as some strangely folded cloth. This is to be expected, since it does not contain any texture. These artifacts are completely invisible in the interpolated views. This confirms that multi-viewpoint interpolation does not require pure geometrically correct correspondences (see section 4.1 and Figure 4.2) opposed to 3-D from stereo applications.

For the face that contains texture and is visible in both images, the quality of both the interpolated images as well as the 3-D model are subjectively good. Since no ground truth data was available for our natural image material, a quantitative evaluation was performed on a subjective basis. When viewing a sequence of the 3-D models as shown in Figure 4.38, we judged that small random noise was present with a size in the order of 5 mm in textureless parts of the face (e.g. cheeks) while areas with rich texture (e.g. nose tip with specular reflections) were stable.

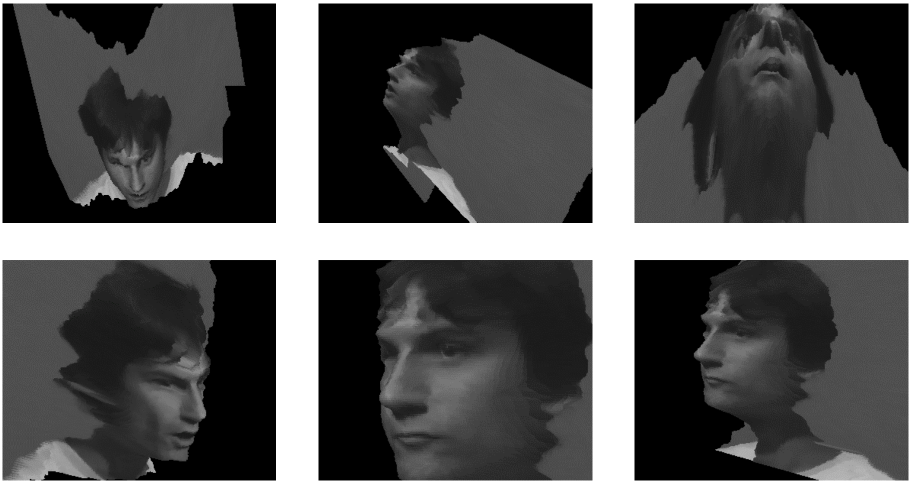


Figure 4.38 The 3-D model obtained from several frames in the sequence, seen from several different viewpoints.

4.9.3 Correspondence estimator for uncalibrated cameras

Whenever the cameras are not in parallel setup, and not calibrated so we cannot rectify their images (see section 2.4.9), we cannot use the algorithm of section 4.9.1. First of all, the correspondence vectors will have a *y* component. But just as important, the camera orientations and zoom factors might differ. This will lead to rotations between the images and scale differences. The smoothness model in section 4.9.1 will regard this as non-smoothness and thus provide a bias towards zero rotation and equal scale. Next we will discuss the changes we must make in the algorithm of section 4.9.1.

Step 1: Definition of fields
Instead of the D_M field, we will use C_A. First, we need to introduce also the *y* component. Secondly, as discussed in section 4.5.1, the C_M field cannot be used in some special cases, e.g. when the cameras differ 180° in orientation.

Step 2: Joint probability model of all fields
We will use the joint and occlusion models from section 4.9.1, but change the luminance and smoothness models. First, the luminance model is given by:

$$U_{\text{luminance}} = \sum_{P_A \in \Lambda_{PA}} \left| I_B\left(P_A + C_A(P_A)\right) - I_A(P_A) \right| \tag{4.44}$$

which only differs from (4.40) in that C_A is used instead of D_M. For the smoothness model we use:

$$U_{\text{smoothness}} = \sum_{P \in \Lambda_{PA}} \left| C_A(P_1) - 2C_A(P) + C_A(P_2) \right|^2 + \left| C_A(P_3) - 2C_A(P) + C_A(P_4) \right|^2 \tag{4.45}$$

Figure 4.39 shows the GRF clique used (see also Appendix C). This is quite different from (4.41). In (4.41), the first derivative of the correspondence field is penalized, while in (4.45) the second derivative is used. This model is invariant for differences in translation, rotation and scale between the *A* and *B* images, as required for this estimator.

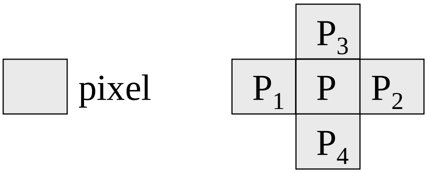


Figure 4.39 Gibbs Random Field clique for the smoothness model on the 2nd derivative.

Step 3: The criterion for the best solution
We will use the MAP criterion, similarly to the algorithm for parallel cameras.

Step 4: The search algorithm
The search algorithm is the same as for the algorithm for parallel image pairs, with only one exception. The perturbation is now a value of a 2-D Gaussian, added to the x and y components of the C_A field.

4.9.4 Experiments

For the correspondence estimator in section 4.9.3, we have performed experiments with synthetic images. The images contained large differences in both rotation (90°) and scale (about 1:2). We have also incorporated curvature effects due to (simulated) lens distortion. The parameter settings for the joint model, SA algorithm and hierarchical approach were the same as for the algorithm for parallel cameras.

Figure 4.40 shows the original images together with interpolated views. Similarly to the parallel algorithm, about 95% of all runs converged to the right solution. From the results we see that the estimator can deal effectively with large differences in rotation and scale. Especially the result with 90° rotation is almost impossible to obtain with a classic correspondence estimation method. Similar results have also not been reported earlier for Bayesian approaches.

The aquarium in the interpolated image is smaller than in the original images. This is no artifact, but due to the pixel-wise interpolation of the left and right images in stead of global rotation. For example, in a sequence of interpolations from the left to the right image, corner points will move in a straight line along the image edge to another corner.

These results open the way for fully automatic and fast correspondence estimation in images from uncalibrated cameras. The correspondence fields found can be used as input for self-calibration methods, enabling fully automatic calibration.

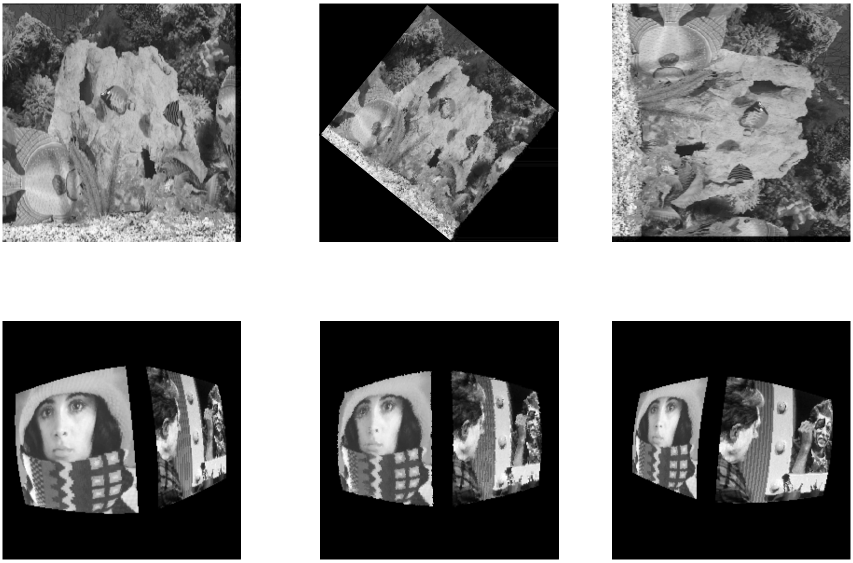


Figure 4.40 Synthetic stereo image pairs (left and right) and interpolated images (middle). The original images contain large rotational differences (top) or scale differences (bottom).

4.10 Conclusions

Correspondence estimation (CE) in a stereo image pair is the most complex and demanding step in the acquisition of 3-D scenes. For this application we require high-resolution (pixel-dense) and high-accuracy correspondence fields. These fields must be calculated in real-time for dynamic scenes. Hence, we require that the computational load of the correspondence estimation is sufficiently low.

A huge amount of literature is devoted to the topic of correspondence estimation, since many other applications exist in which correspondence estimation plays an important role. These include MPEG-4 object-based video coding, multi-viewpoint image generation, camera calibration, structure from motion and 3-D from stereo applications. All of these applications require geometric correspondences. Such a correspondence represents a 3-D scene point, while a photometric correspondence just represents photometric similarity between image points. The high-resolution and high-accuracy estimation of geometric correspondences requires complex dense field models. These reflect the prior model of the scene to be acquired. Such a model can only be designed on a heuristic basis, which explains the great diversity in CE algorithms. Since a model in general does not have causal properties (along the image axes), the estimation of all correspondences should in principle be done simultaneously. If the correspondences are estimated one by one for the whole image, implicitly a causality constraint is imposed on the dependencies between the correspondences in the model.

The classic approaches to correspondence estimation, feature detection and matching, block matching, pel-recursive algorithms and optical-flow methods are not suitable for our application. They do not yield a pixel-dense correspondence field, they fail to estimate the entire field simultaneously, or they are only successful for stereo images with very small baseline (images are almost the same).

More recently, several promising algorithms have been developed that are suited for our application. These employ the Bayesian approach, which uses explicit probability models of the images, the correspondence fields and their segmentation. This increases the portability and adaptability of algorithms among different applications and different designers. The ingredients of the models can be categorized in photometric and geometric models. Photometric models include image luminance and its discontinuities in relation to those of the correspondence field. Geometric models, needed for geometric correspondence estimation, currently include a priori models for occlusions, and continuity and smoothness of correspondence.

We recognized four distinct steps in the Bayesian approach. First, all variable fields to be estimated are clearly defined. Secondly, a joint probabilistic model is designed for all variables of all fields. Then, in the third step, a criterion (e.g. MAP) is adopted to define the best solution mathematically. Finally, in the fourth step some search algorithm estimates this solution.

In the field definition and modeling steps, we reviewed many fields and models found in literature and categorized them using a uniform notation. We focused at Gibbs and Markov Random Field (GRF, MRF) models that are especially suited for the Bayesian approach. Further, we reviewed several special fields and models that have already appeared in CE algorithms, but have not yet been used for the application of dense geometric fields and provide areas for future research.

We found that several models for interacting fields cannot be combined using the Bayes rule, opposed to some false attempts in literature. A way to circumvent this is to refrain from using this rule in the probability domain and combine the models in the energy domain. This is at the cost of explicitness of the modeling, but an advantage is that the joint model can be synthesized more freely and can include more submodels and constraints.

In the solution definition step, we concluded that the Maximum A Posteriori (MAP) and Mean Field (MF) criteria are applicable for CE purposes. The Maximum Likelihood (ML) criterion is not. Approaches in literature that claim to use the ML criterion always turn out to use the MAP criterion implicitly after having a close look on the algorithm.

In the search algorithm step, we reviewed many algorithms such as downhill descent, Simulated Annealing (SA) and Dynamic Programming (DP) techniques. We found that the combination of the cooling schedule in SA algorithms and the popular hierarchical speedup method results in a simpler version of the SA algorithm, which in addition has far better computational aspects.

We have designed new correspondence estimators based on the requirements we have set for our application and the findings in our review. Two algorithms were derived: one for parallel image pairs and one for general pairs from uncalibrated stereo cameras. We designed both completely using the Bayesian approach, with MRF models, the MAP criterion and search algorithms on the basis of simulated annealing (SA) in a hierarchical fashion. Both algorithms had run times in the order of 20 seconds, which is orders of magnitude faster than other results from literature. This is ascribed fully to the combination of the simpler SA algorithm and the hierarchical approach. The low computational aspects of the algorithms show that the Bayesian methods with MRF models and SA search algorithms are no longer only in the domain of research, but will also become available soon in practical real-time applications. Our implementation is still a factor of about 500 slower than real-time, but a hardware implementation may use the fact that MRF algorithms can be implemented in massive parallel computational structures.

With the algorithm for parallel image pairs, we obtained high-quality correspondence fields. They were subjectively evaluated by using them for image interpolation and 3-D model generation. High-quality 3-D models were obtained from stereo images typical for a 3-D videoconferencing application. With the algorithm for uncalibrated cameras, we have performed experiments with synthetic stereo images that contain large rotational and scale differences. High-quality correspondence fields were obtained that show the robustness of the algorithm against these differences to an extent not reported in literature before.

In future research related directly to our algorithms, the introduction of a temporal consistency constraint might eliminate the few spurious results as in Figure 4.37. Additionally, it would increase the accuracy by lowering the random noise on the 3-D models. As an application for our algorithms, the fields estimated with them are still to be used for self-calibration of cameras.

A big issue to be solved is the evaluation of our and similar algorithms with objective and meaningful quality measures. The measures we used were subjective, enabling rough evaluation but no quantitative comparison with other methods. In literature, the majority of CE algorithms are used for video coding, where PSNR and coding efficiency are valid quality measures. In the domain of 3-D scene acquisition from stereo, a different measure is needed.

Chapter 5

3-D scene visualization on stereo displays

5.1 Introduction

This chapter deals with the 3-D scene visualization part of the adaptive multi-viewpoint system defined in the introduction of this thesis and shown in Figure 5.1. The task of the visualization part is to provide the viewer with a stereo image pair in such a way that the 3-D scene is visualized correctly. The stereo image pair is generated on the basis of the incoming 3-D scene model. To allow the viewer to observe the scene from his own angle of interest, it is necessary to adapt the generated images continuously to follow the position of the viewer's eyes. The stereo display shows the two images of the stereo pair to the viewer's left and right eye separately, either by using special glasses or by means of an autostereoscopic display that does not impose any eye wear on the viewer.

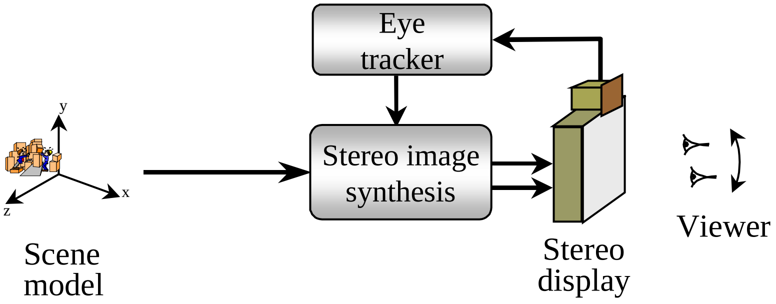


Figure 5.1 The scene visualization part of the multi-viewpoint system.

Two signal-processing tasks are involved in the visualization part:

- 3-D tracking of the viewer's eyes.
- Generation of the correct stereo image pair on the basis of a 3-D scene model and the eye coordinates.

For the stereo display, products are commercially available, e.g. [Hein, Phil] or PC monitors in stereo mode. Eye trackers are commercially available that operate on the basis of markers on the viewer, e.g. [Orig]. In research, also non-invasive trackers are available on the basis of eye recognition in images taken from the viewer, see e.g. [Rede97e]. We used the tracker from [Orig] and a high resolution PC display in stereo mode in combination with professional LCD shutter glasses [Ster].

Image synthesis, for our application multi-viewpoint image synthesis, is currently an important topic in computer vision research. Many algorithms are present that generate images from virtual cameras, on the basis of images taken by real cameras. The virtual cameras are either positioned and orientated arbitrarily [Faug96, Fuji96, Levo96] or restricted to a position in between the two real cameras of a stereo pair. In the latter case, image interpolation can be used to synthesize the images [Chup94, Ohm97, Seit95, Veig96]. Images that result from all these algorithms can in principle be acquired by appropriately positioned real cameras. However, such images, if they are shown on a stereo display, cannot visualize the scene without geometrical errors. A correction is needed where the entire image is shifted a number of pixels in the x and y directions. This has been shown for stereo systems [Grin94, Kutk94] and for multi-viewpoint systems on the basis of image interpolation [Kang96, Pano95, Rede97f]. In both cases, only the x component of the shift is involved. In [Pasm97], a similar result is obtained for a mechanical fixed multi-viewpoint system on the basis of X-ray images. In this chapter we examine the general case for adaptive multi-viewpoint systems. This investigation is based on [Rede97b, Rede00].

Whenever multi-viewpoint images are constructed with the general algorithm incorporating the aforementioned shift, the 3-D scene visualization might still suffer from geometric errors for several reasons. Clearly, the resolution of the display is finite, which limits the resolution of the visualized scene. Further, any practical eye tracker has a finite accuracy, resulting in images that are slightly different from the ideal ones. To our knowledge, an analysis of these effects on the scene visualization has not yet been performed for multi-viewpoint systems. Previous analyses that very slightly resemble such an analysis can be found in [Vais99] for augmented reality systems with head-mounted displays, and in the area of camera calibration [Tsai87], where triangulation errors of calibration points in 3-D space (similar to our scene points) are derived on the basis of 2-D image feature locations (similar to points in the multi-viewpoint images). We will examine the effects of eye tracker errors in detail, based on [Rede00].

Further, in any practical system there is a delay between the eye-tracker measurement, the synthesis of the multi-viewpoint images and finally, the observation of the images by the viewer. In augmented reality systems, where the viewer observes the visualized scene together with the real environment using a head-mounted display, the latency errors are more annoying since they are present in the virtual part of the scene, but not in the real scene. In this area the effects of latency have been studied and are counteracted by special low-latency rendering techniques [Pasm99]. We will study the effects of latency in multi-viewpoint systems, based on our results for eye tracker errors.

In section 5.2, we will consider the general case for the generation of multi-viewpoint images. In section 5.3 we analyze the geometrical errors in the visualization due to eye-tracking errors in general. The other two causes of visualization errors, the resolution of the display and the rendering latency, are described as well using the former analysis. In section 5.4 we validate our theoretical findings by an extensive experiment with human subjects. Finally, section 5.5 concludes the chapter. In all sections we will use the notation from Appendix A.

5.2 Generation of multi-viewpoint images

In section 5.2.1, we will derive the correct algorithm for the generation of the multi-viewpoint images. This algorithm provides scene visualization without any geometrical errors, and is simple at the same time. However, a few practical challenges exist for through-the-window systems due to the display size. In section 5.2.2 we will examine these challenges and provide a solution by means of manipulating the visualized scene manually.

5.2.1 Image synthesis for viewpoint-adaptive visualization

Figure 5.2 shows the stereo display, with in the center the display reference frame O_D, with the meter as unit (not pixels). We assume that the display is perfectly flat and thus occupies the $z_D = 0$ plane. The O_I reference frame is the pixel frame of the display with continuous coordinates x_I and y_I (see Chapter 2 and Appendix A for our notation).

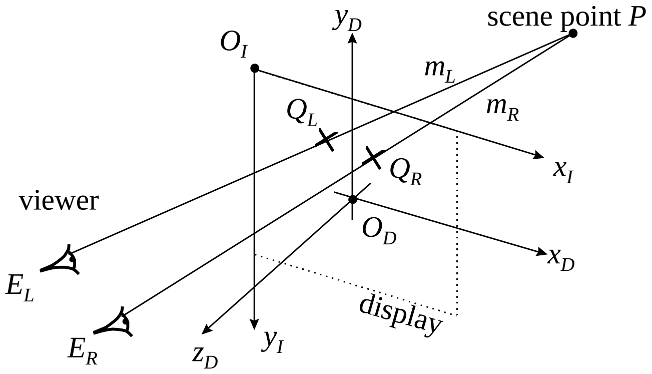


Figure 5.2 Viewpoint-adaptive visualization of a 3-D scene point.

For each scene in the model, we construct the two light rays m_L and m_R, which leave from this point and enter the viewer's eyes. These rays go through the pupils E_L and E_R (optical centers of the eyes). To construct m_L and m_R, we only need the exact position of the pupils. Since no information is needed about the retinas of the eyes, this application does not need information about the gaze direction. This construction can be done independently for each eye and thus we will discuss only one by omitting the L or R subscripts. After some basic geometrical calculations, we find for the coordinates of Q:

$$\begin{bmatrix} Q^{x_D} \\ Q^{y_D} \end{bmatrix} = \frac{1}{P^{z_D} - E^{z_D}} \begin{bmatrix} P^{z_D} E^{x_D} - E^{z_D} P^{x_D} \\ P^{z_D} E^{y_D} - E^{z_D} P^{y_D} \end{bmatrix} \tag{5.1}$$

The superscripts denote the coordinates in the display frame O_D, see Appendix A. We always have $E^{z_D} > 0$ and $Q^{z_D} = 0$. For points behind the display we have $P^{z_D} < 0$ and for the points in front of the display have $P^{z_D} > 0$. If the pixels of the display have size s_{dx} and s_{dy}, then the (continuous) image coordinates x_I, y_I of Q are:

$$\begin{bmatrix} Q^{x_I} \\ Q^{y_I} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} N_x \\ N_y \end{bmatrix} + \begin{bmatrix} Q^{x_D} / s_{dx} \\ -Q^{y_D} / s_{dy} \end{bmatrix} \tag{5.2}$$

Here the N_x and N_y denote the size of the display in pixels. If we e.g. round off the continuous image coordinates to integer pixel coordinates, we may copy the luminance (and color) of scene point P from the scene model into the left and right image at the positions Q_L and Q_R, respectively. For Lambertian (diffuse radiating) scene surfaces, the luminances at Q_L and Q_R are the same. For specular reflecting surfaces, they may differ. If this construction is done for all scene points P and for both eyes, we obtain a stereo pair of multi-viewpoint images that enable visualization of the scene without any geometrical distortion.

The construction (5.1) and (5.2) for a single eye is very similar to, but not exactly the same as image formation in a normal camera. The eye and the display in our construction play the role of the optical center and the projection plane in a camera. The major difference is the position of the optical center: in a normal camera, it is on (or very near to) the z axis, but in Figure 5.2 it may be at any position. This has been recognized for stereo systems [Grin94, Kutk94], and multi-viewpoint systems on the basis of interpolation [Konr99, Ohm98] and extrapolation [Rede97b].

Generally, three phenomena arise during the construction of the multi-viewpoint images:

- Q^{x_I} and Q^{y_I} are not integers.
- Multiple scene points project to the same pixel.
- A pixel never gets assigned a luminance, since no scene point is projected to it.

In the area of rendering normal 2-D images from 3-D scene models, which is standard computer graphics, these phenomena have been dealt with extensively. However, we will examine their geometric effects in the context of the 3-D visualization in multi-viewpoint systems.

In the first case, geometrical errors due to pixel discretization can be made arbitrarily small by first constructing a high-resolution image followed by extracting a normal-resolution image by low-pass filtering and subsampling. This produces at most slight loss of detail due to blurring.

The multiple projection case is exactly the same as in normal image formation. We assume that the original scene consists of opaque surfaces only. Then we project the scene point that is closest to the viewer or, equivalently, has largest P^{z_D}. This conforms to the fact that opaque objects occlude each other and that only the closest object remains visible.

In the last case, “holes” are visible in the displayed scene. Even when the scene is captured by a method with a large number of cameras and the scene model is very complex, the viewer may always take a viewpoint for which the scene model does not contain enough information. This is inevitable. However, in general the scene model should contain those parts of a scene the viewer is interested in and then the holes are automatically avoided.

Much smaller but more annoying holes arise in the following situation. Scene models exist that contain only of a cloud of points *P*, without any surfaces defined between them (e.g. image-based models as used in Chapter 6). In principle, such a scene is not visible at all since points are infinitely small and thus have zero area. If all points are projected to pixels, they implicitly “gain” some surface and they are visualized. This may lead to a large number of holes with sizes in the order of one or a few pixels. Such holes are very annoying, but can be avoided easily by interpolation of nearby pixels [Rede97b].

5.2.2 Manual scene manipulation

If large changes in viewing angle are needed, e.g. to see the back of the scene, in real-life we can walk around objects. In “through-the-window” based video systems, this is not possible. The viewing position and direction are constrained, since the lines of sight from viewer to scene must always intersect the display.

Figure 5.3 shows a worst-case example, in which a viewer is looking at a scene that is orders of magnitude larger than the display. Walking around the scene means that the lines of sight do no longer intersect the display and thus, the scene visualization is lost.

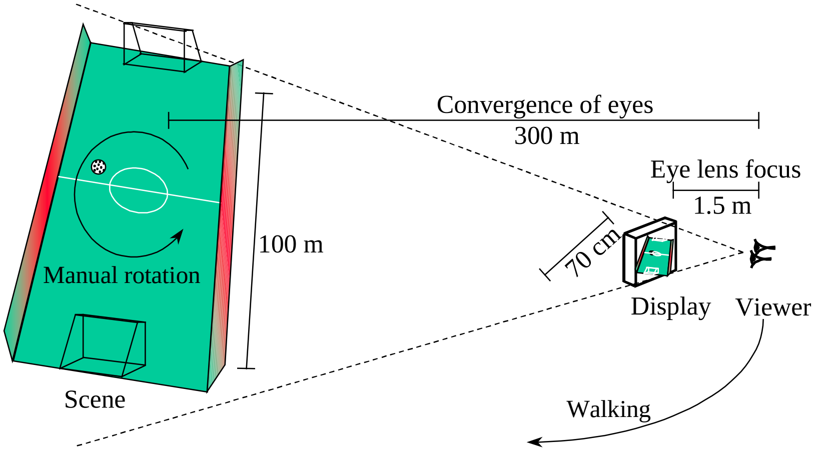


Figure 5.3 Walking around a large visualized scene cannot be done, but it can be simulated by manually rotating the scene.

This can be solved by allowing for manual rotation, translation and scaling of the scene model with respect to the display:

$$P^{\sigma_D} = SV^{\sigma_D}_{\sigma_{model}} P^{\sigma_{model}} + O^{\sigma_D}_{model} \tag{5.3}$$

Here S is a scale factor, while $V_{\sigma_{model}}^{\sigma_D}$ is a rotation matrix and $O_{model}^{\sigma_D}$ is a translation vector as defined in Appendix A. We can now simulate walking around in two ways. The most direct method is to allow the viewer to rotate the scene model manually. The second method is to make the scene smaller by manually scaling it to the size of the display, and then to translate it to the center of the display. After that, head movement can again be used, which now provides almost 180° degrees of viewing angle. Although the visualization is now correct only up to a scale factor, the increased motion parallax possibility and scene overview features seem very useful. Additionally, all scene objects now appear approximately centered in the display. Then, on average, the eyes will converge to the depth of the display. For displays that do not provide the eye lens accommodation cue, this minimizes the accommodation-convergence conflict as described in Chapter 1 (in Figure 5.3 this conflict is quite large).

5.3 Geometrical errors

If multi-viewpoint images are generated as discussed in section 5.2, several causes remain that may contribute to geometrical errors in the scene visualization. The display has finite resolution, it may not be perfectly flat due to intended curvature or (small) deformations, and the thickness and type of the display front may cause light rays to refract slightly before travelling into the air. Further, the measurements from the eye-tracker may not be exact. This can be because the accuracy of the tracker itself is finite, or due to discretization of the measured eye positions by a practical implementation of the visualization algorithm. Also the calibration between display and tracker (transformation of coordinates from tracker to display) may not be correct. When rendering latency is present in the system, the images are no longer synthesized based on tracker data from the present, but on data from some moment in the past.

In this section we will first analyze the effect of general eye-tracking errors on the visualization. Secondly, we will relate these results to the resolution of the human eye and display. A simple bound on the allowed eye-tracking errors will be formulated, below which the eye-tracking errors result in unobservable visualization errors. Finally, a bound will be derived for the maximum system latency. In these sections we assume that the calibration of display and tracker is performed without error, and that the display is perfectly flat and does not refract the light rays it emits.

5.3.1 Eye-tracking errors

Eye-tracking errors cause two effects in the visualization of each scene point, illustrated quite exaggeratedly in Figure 5.4:

- The observed scene point $\hat{P}$ does not have the same position as P.
- The two lines of sight do not intersect, but cross with a minimal distance $|\Delta D|$ in between.

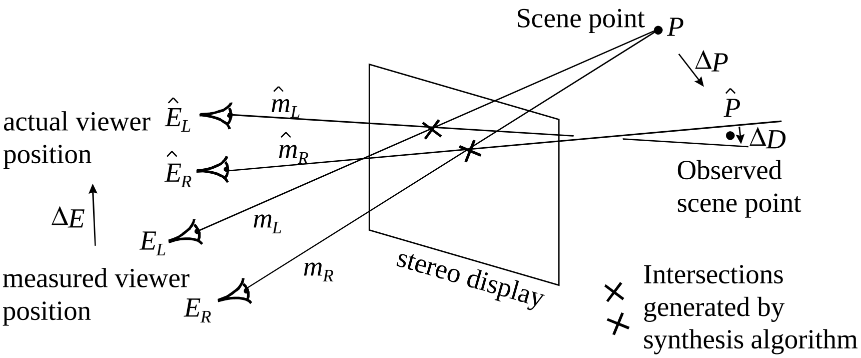


Figure 5.4 Two effects of eye-tracking errors (compare with Figure 5.2).

The first effect produces a geometrically distorted scene. The second effect never occurs when looking at real objects. It may be partly responsible for headaches caused by stereo systems, although we found no research related to this in literature. We assume that if $|\Delta D|$ is small, the brain can still merge the two lines of sight and that it observes the point $\hat{P}$ centered on the vector ΔD. The vector ΔD is similar to the intersection error defined at the end of section 2.5.4, which is now encountered by the human brain.

In Figure 5.4, all points without hat refer to the situation as seen by the visualization algorithm. All points with hat refer to the actual situation. The error ΔP is defined as the vector from P to $\hat{P}$. The ΔD is defined as the smallest vector that can be found from a point on $\hat{m}_R$ to a point on $\hat{m}_L$, similar to the definition in section 2.5.4. The errors ΔE_L, ΔE_R are defined as the vectors from E_L, E_R to $\hat{E}_L$, $\hat{E}_R$, respectively.

The ΔP and ΔD are non-linear functions of ΔE_L, ΔE_R, E_L, E_R and P. For the analysis, we introduce (see Figure 5.5) head position H, located midway between the eyes, a line m_H and display intersection point Q_H. Further we introduce the inter-eye distance d_{eye}, the binocular viewing angle α and plane Φ, which contains the lines m_L, m_R and m_H and consequently, the points E_L, E_R, H, Q_L, Q_R, Q_H and P. Finally, we introduce three new reference frames O_L, O_R and O_H, whose origins are located in the display at Q_L, Q_R and Q_H, respectively.

The y_L, y_R and y_H axes are all defined perpendicular to the plane Φ, which yields $y_L = y_R = y_H$. The z_L, z_R and z_H axes are contained in the lines m_L, m_R and m_H, respectively, and point in the direction of the viewer. For right handed references frames, the three x axes are then fixed and lie within the plane Φ, orthogonal to the lines of sight m_L, m_R and m_H as shown in Fig. 5.5.

We will decompose the tracking errors ΔE_L, ΔE_R in O_L and O_R, respectively. Assuming that the six components are small, we can linearize their effects, analyze these separately and obtain the total result by superposition. For the coordinates of the eye positions E_L, E_R, the scene point P and the components of visualization errors ΔP and ΔD, we will use whatever frame is most appropriate. The superposition of all errors requires in the end that all coordinates are transformed to the same reference frame, which will be O_H.

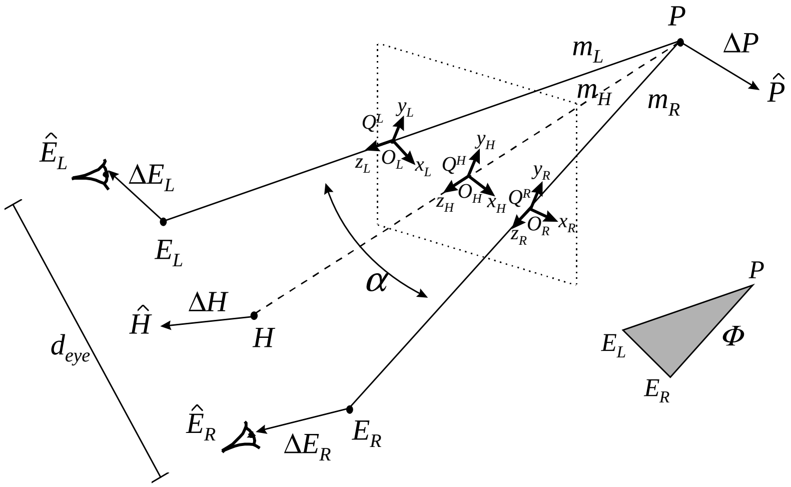


Figure 5.5 Head position and reference frames for the analysis of eye-tracking errors.

Figure 5.6 shows the three components of the left eye error and their effects. The ΔE_L in the $+x_L$ direction produces a small ΔP within Φ, in the direction of $-z_R$. No ΔD is produced.

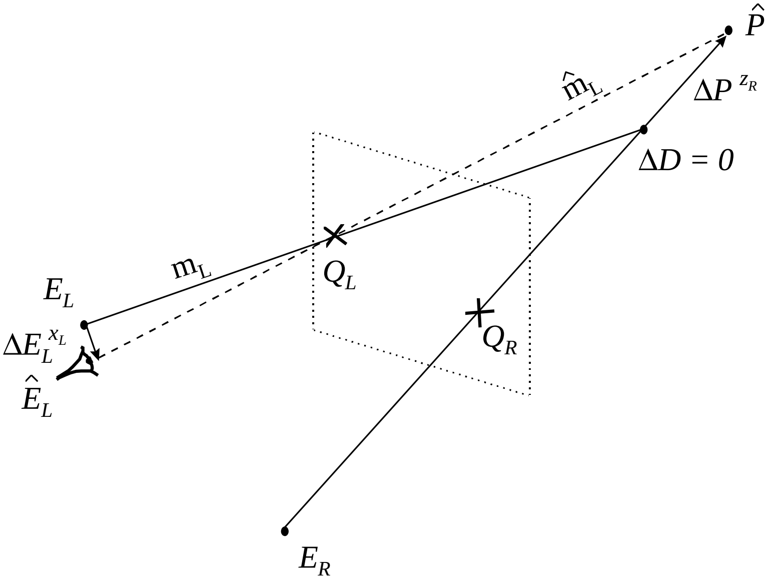


Figure 5.6a The effects of left-eye tracking errors in the x_L direction.

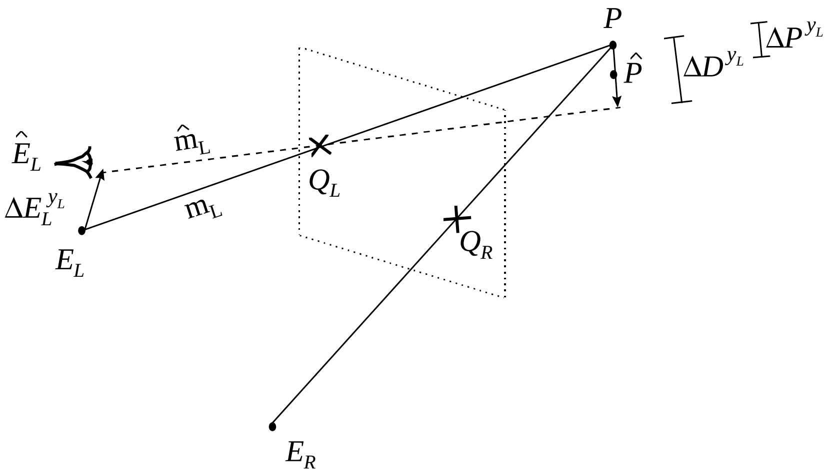


Figure 5.6b The effects of left-eye tracking errors in the y_L direction.

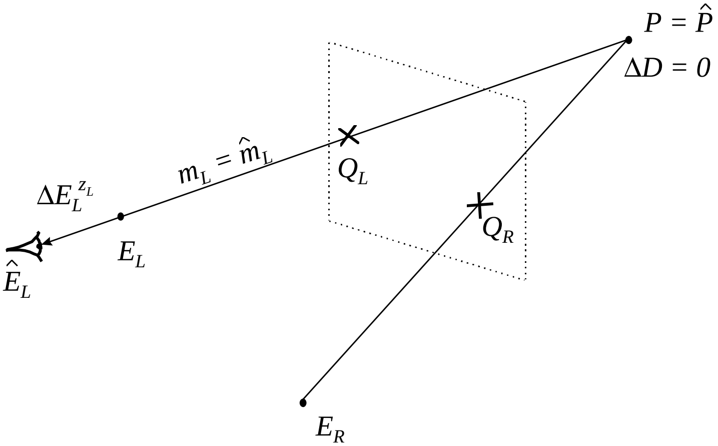


Figure 5.6c The effects of left-eye tracking errors in the z_L direction.

The size of ΔP is related to α and to the quotient of the lengths of m_L in front of and behind the display (viewer-display and display-point distance respectively). After some calculation we find:

$$\Delta P^{z_R} = \frac{1}{\sin\alpha}\frac{P^{z_L}}{E_L^{z_L}}\Delta E_L^{x_L} \tag{5.4}$$

Small ΔE_L in the $+y_L$ direction produce small ΔP and ΔD perpendicular to Φ in the direction of $-y_L$ (or $-y_R$ or $-y_H$). Their sizes are:

$$\Delta D^{y_L} = \frac{P^{z_L}}{E_L^{z_L}} \Delta E_L^{y_L} \tag{5.5}$$

$$\Delta P^{y_L} = \frac{1}{2} \frac{P^{z_L}}{E_L^{z_L}} \Delta E_L^{y_L} \tag{5.6}$$

Any ΔE_L in the z_L direction does not effect the visualization at all, since it has no effect on the line of sight m_L.. Similarly, we find for the visualization errors due to the right eye errors ΔE_R:

$$\Delta P^{z_L} = -\frac{1}{\sin\alpha} \frac{P^{z_R}}{E_R^{z_R}} \Delta E_R^{x_R} \tag{5.7}$$

$$\Delta D^{y_R} = -\frac{P^{z_R}}{E_R^{z_R}} \Delta E_R^{y_R} \tag{5.8}$$

$$\Delta P^{y_R} = \frac{1}{2} \frac{P^{z_R}}{E_R^{z_R}} \Delta E_R^{y_R} \tag{5.9}$$

To allow for superposition of the visualization errors, we must transform all coordinates on the left-hand sides of (5.4) to (5.9) from the O_L and O_R frames to those of O_H. Figure 5.7 shows the relation between the coordinates.

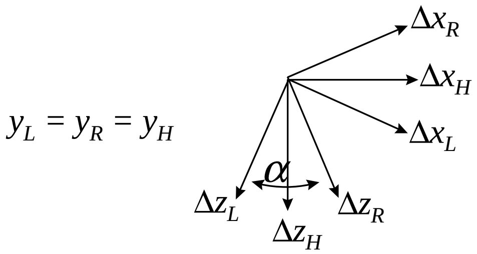


Figure 5.7 O_L, O_R and O_H coordinates.

We assume that the O_H coordinates are centered in between the O_L and O_R coordinates. If the viewer looks at P and his nose points towards P, the lines m_L and m_R have equal length and this holds true. For other head orientations this is generally not true. For small α, the deviations are very small and can be neglected. For α we find:

$$\alpha \le \tan\alpha \approx \frac{d_{eye}}{H^{z_H} - P^{z_H}} \tag{5.10}$$

We approximate the inter-eye distance by $d_{eye} \approx 7$ cm. Then, if the distance between the viewer and point P is in the order of 1 m or larger, we find $\alpha < 0.07$ rad or smaller. Thus, we are allowed to use the symmetric situation in Figure 5.7 and find, for example:

$$\begin{bmatrix} \Delta x_H \\ \Delta z_H \end{bmatrix} = \begin{bmatrix} \cos \tfrac{1}{2}\alpha & -\sin \tfrac{1}{2}\alpha \\ \sin \tfrac{1}{2}\alpha & \cos \tfrac{1}{2}\alpha \end{bmatrix} \begin{bmatrix} \Delta x_L \\ \Delta z_L \end{bmatrix} \tag{5.11}$$

For small α, we can apply:

$$\begin{bmatrix} \sin\alpha \\ \cos\alpha \end{bmatrix} \approx \begin{bmatrix} \alpha \\ 1 \end{bmatrix} \tag{5.12}$$

Further, we use the approximation:

$$\frac{P^{z_L}}{E_L^{z_L}} \approx \frac{P^{z_R}}{E_R^{z_R}} \approx \frac{P^{z_H}}{H^{z_H}} \tag{5.13}$$

If we apply (5.11) and similar rotational transforms, (5.12), and (5.13) to both the left-hand and right-hand sides of (5.4) to (5.9), we obtain after superposition:

$$\Delta D^{y_H} = \frac{P^{z_H}}{H^{z_H}} \left(\Delta E_L^{y_H} - \Delta E_R^{y_H} \right) \tag{5.14}$$

and

$$\begin{bmatrix} \Delta P^{x_H} \\ \Delta P^{y_H} \\ \Delta P^{z_H} \end{bmatrix} = \frac{P^{z_H}}{H^{z_H}} \frac{1}{2} \left\{ \begin{bmatrix} \Delta E_L^{x_H} + \Delta E_R^{x_H} \\ \Delta E_L^{y_H} + \Delta E_R^{y_H} \\ \Delta E_L^{z_H} + \Delta E_R^{z_H} \end{bmatrix} + \begin{bmatrix} \tfrac{\alpha}{2}\left(\Delta E_L^{z_H} - \Delta E_R^{z_H} \right) \\ 0 \\ \tfrac{2}{\alpha}\left(\Delta E_L^{x_H} - \Delta E_R^{x_H} \right) \end{bmatrix} \right\} \tag{5.15}$$

We assume that the eye-tracking errors have uncorrelated, zero mean components, with variances that are possibly different in x_H, y_H and z_H directions but equal for left and right eyes. We denote the eye-tracking errors by $\sigma_{\Delta E;x}$, $\sigma_{\Delta E;y}$ and $\sigma_{\Delta E;z}$. Then all sums on the right hand sides of (5.14) and (5.15) are uncorrelated with respect to each other, as well as the left hand components. This results in:

$$\begin{bmatrix} \sigma^2_{\Delta D;y_H} \\ \sigma^2_{\Delta P;x_H} \\ \sigma^2_{\Delta P;y_H} \\ \sigma^2_{\Delta P;z_H} \end{bmatrix} = \frac{P^2_{z_H}}{H^2_{z_H}} \left\{ \begin{bmatrix} 2\sigma^2_{\Delta E;y} \\ \tfrac{1}{2}\sigma^2_{\Delta E;x} \\ \tfrac{1}{2}\sigma^2_{\Delta E;y} \\ \tfrac{1}{2}\sigma^2_{\Delta E;z} \end{bmatrix} + \begin{bmatrix} 0 \\ \tfrac{\alpha^2}{8}\sigma^2_{\Delta E;z} \\ 0 \\ \tfrac{2}{\alpha^2}\sigma^2_{\Delta E;x} \end{bmatrix} \right\} \tag{5.16}$$

As just deduced, α is in the order of 0.1 rad. For eye trackers with isotropic errors, (5.16) can be approximated to:

$$\begin{bmatrix} \sigma_{\Delta D;y_H} \\ \sigma_{\Delta P;x_H} \\ \sigma_{\Delta P;y_H} \\ \sigma_{\Delta P;z_H} \end{bmatrix} = \sqrt{2}\frac{\left|P^{z_H}\right|}{H^{z_H}} \begin{bmatrix} 1 \\ 1/2 \\ 1/2 \\ 1/\alpha \end{bmatrix} \sigma_{\Delta E} \tag{5.17}$$

Where $\sigma_{\Delta E}$ refers to all spatial directions. Many tracking algorithms produce spatially anisotropic errors. For the commercial tracker from [Orig] and the experimental tracker in [Rede97e], $\sigma_{\Delta E;x}$ and $\sigma_{\Delta E;y}$ have equal magnitude, but the errors in the z direction $\sigma_{\Delta E;z}$ are about 5 times larger. However, if $\sigma_{\Delta E;z} < 2/\alpha\, \sigma_{\Delta E;x}$, then (5.16) still approximates:

$$\begin{bmatrix} \sigma_{\Delta D;y_H} \\ \sigma_{\Delta P;x_H} \\ \sigma_{\Delta P;y_H} \\ \sigma_{\Delta P;z_H} \end{bmatrix} = \sqrt{2}\frac{\left|P^{z_H}\right|}{H^{z_H}} \begin{bmatrix} \sigma_{\Delta E;y} \\ 1/2\,\sigma_{\Delta E;x} \\ 1/2\,\sigma_{\Delta E;y} \\ 1/\alpha\,\sigma_{\Delta E;x} \end{bmatrix} \tag{5.18}$$

For the aforementioned trackers, this equals (5.17), since their errors are isotropic in the x_H and y_H directions.

For complexity reasons, multi-viewpoint systems may adapt only to one or two dimensions of viewer movement. For example, in a system based on image interpolation, the system adapts only to viewer motion in the x_H direction. The system assumes that the viewer remains at some specific y_H and z_H position. This system functions the same as a fully adaptive system that suffers from severe eye tracking errors, namely the measured y_H and z_H positions are fixed. All the movements of the viewer around the prescribed y and z position then define both $\sigma_{\Delta E;y}$ and $\sigma_{\Delta E;z}$, which will be substantially larger than $\sigma_{\Delta E;x}$. In these cases, we do have $\sigma_{\Delta E;z} > 2/\alpha\, \sigma_{\Delta E;x}$, and (5.16) can be approximated to:

$$\begin{bmatrix} \sigma_{\Delta D;y_H} \\ \sigma_{\Delta P;x_H} \\ \sigma_{\Delta P;y_H} \\ \sigma_{\Delta P;z_H} \end{bmatrix} = \sqrt{2}\frac{\left|P^{z_H}\right|}{H^{z_H}} \begin{bmatrix} \sigma_{\Delta E;y} \\ \alpha/4\,\sigma_{\Delta E;z} \\ 1/2\,\sigma_{\Delta E;y} \\ 1/2\,\sigma_{\Delta E;z} \end{bmatrix} \tag{5.19}$$

Also, the tracking errors may be effectively enlarged by discretization of the measured eye positions by a practical implementation of the rendering algorithm. In Chapter 6 we will deal with such multi-viewpoint systems as specific cases of the generic algorithm defined in section 5.2.

5.3.2 Resolution of the human eye and the display

In this section we will derive a bound below which eye-tracking errors yield unobservable visualization errors.

A rule of thumb for all displays is that we must keep a distance from the display of a few times its size. If we observe this rule, the system resolution is limited by the eye instead of by the display. Roughly speaking, the spatial resolution of the eye is below 1/30th of a degree [Wand95], which is $\alpha_{eye} \approx 6x10^{-4}$ rad.

Figure 5.8 shows a visualized point *P* and two cones, each with its apex at an eye pupil and both aimed at *P*. These cones represent the light rays that are used to observe point *P*. The angular width of the cones is a_{eye}, representing one unit of spatial resolution of each of the eyes.

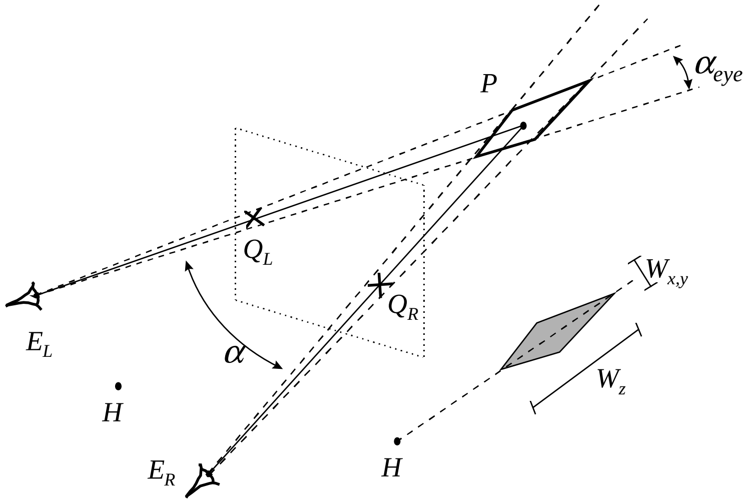


Figure 5.8 The resolution of the eye.

All ΔP that shift *P* to another position within the intersection of both cones are not visible. Since $\alpha_{eye} << \alpha$, the width of each cone around the intersection can be assumed constant. Applying (5.13) we find for the width $W_{x,y}$ of both cones in the x_H and y_H direction:

$$W_{x,y} \approx \alpha_{eye}\left(H^{z_H} - P^{z_H}\right) \tag{5.20}$$

For small α, the width of the intersection equals that of the cones. For the length W_z of the intersection we find:

$$W_z \approx \frac{2W_{x,y}}{\tan\alpha} \approx \frac{2W_{x,y}}{\alpha} \tag{5.21}$$

The requirement that ΔP lies somewhere within the intersections of the cones can then be formulated as:

$$\begin{bmatrix} \left|\Delta P^{x_H}\right| \\ \left|\Delta P^{y_H}\right| \\ \left|\Delta P^{z_H}\right| \end{bmatrix} < \alpha_{eye}\left(H^{z_H} - P^{z_H}\right) \begin{bmatrix} 1/2 \\ 1/2 \\ 1/\alpha \end{bmatrix} \tag{5.22}$$

The σ corresponding to uniformly distributed errors within some interval, is equal to the length of the interval divided by √12. In this way, we can rewrite (5.22) as:

$$\begin{bmatrix} \sigma_{\Delta P;x_H} \\ \sigma_{\Delta P;y_H} \\ \sigma_{\Delta P;z_H} \end{bmatrix} < \frac{\alpha_{eye}}{\sqrt{12}}\left(H^{z_H} - P^{z_H}\right) \begin{bmatrix} 1 \\ 1 \\ 2/\alpha \end{bmatrix} \tag{5.23}$$

This enables us to combine (5.23) with (5.18), (5.19) and (5.10) into:

$$\begin{bmatrix} \sigma_{\Delta E;x} \\ \sigma_{\Delta E;y} \\ \sigma_{\Delta E;z} \end{bmatrix} < \frac{\alpha_{eye} H^{z_H}}{\sqrt{6}\left|P^{z_H}\right|}\left(H^{z_H} - P^{z_H}\right) \begin{bmatrix} 1 \\ 1 \\ 2\left(H^{z_H} - P^{z_H}\right) / d_{eye} \end{bmatrix} \tag{5.24}$$

For points far behind the display, we have $P^{z_H} \rightarrow -\infty$ and find:

$$\begin{bmatrix} \sigma_{\Delta E;x} \\ \sigma_{\Delta E;y} \\ \sigma_{\Delta E;z} \end{bmatrix} < \frac{\alpha_{eye} H^{z_H}}{\sqrt{6}} \begin{bmatrix} 1 \\ 1 \\ \infty \end{bmatrix} \tag{5.25}$$

For head positions in the order of 1 m from the display, visualization errors remain unobservable when the eye-tracking errors are smaller than 0.2 mm in the *x* and *y* directions. The eye-tracker error in the *z* direction is not restricted.

As discussed in section 5.2.2, it is advantageous to visualize the scene in the center of the display. Then P^{z_H} is small compared to H^{z_H} and we obtain:

$$\begin{bmatrix} \sigma_{\Delta E;x} \\ \sigma_{\Delta E;y} \\ \sigma_{\Delta E;z} \end{bmatrix} < \frac{\alpha_{eye} \left|H^{z_H}\right|^2}{\sqrt{6}\left|P^{z_H}\right|} \begin{bmatrix} 1 \\ 1 \\ 2H^{z_H} / d_{eye} \end{bmatrix} \tag{5.26}$$

It must be noted that each point in the scene model has its own, unique O_H reference frame and coordinates. Therefore, the x_H , y_H and z_H coordinates in (5.26) refer to different directions for different points. A simple and absolute safe bound can be obtained by taking the minimum of the three bounds over all points. In this case (5.26) reduces to an isotropic bound that is the same for all points:

$$\sigma_{\Delta E} < \frac{\alpha_{eye} \left| H^{z_H} \right|^2}{\sqrt{6} \left| P^{z_H} \right|_{\max}} \tag{5.27}$$

This bound is quite over-restrictive for the z_H component of the eye-tracking error for all points, and over-restrictive for all components of points closer to the display than $\left| P^{z_H} \right|_{\max}$. For scene points about $\left| P^{z_H} \right|_{\max}$ = 15 cm around the display (for example a teleconferencing application with human heads as scene), and a viewer distance of about 1 m, we obtain $\sigma_{\Delta E} < 1$ mm. Via (5.26) we observe that the actual allowed eye-tracking error in the z direction is about 30 times larger.

5.3.3 Rendering latency

We will derive a simple and safe bound for rendering latency, below which artifacts cannot be observed. A rendering latency of t seconds has only effects when the viewer is changing his viewpoint. In that case, the images shown on the display correspond to a viewer position from t seconds in the past. Effectively, the latency produces errors equivalent to those of eye-tracking errors with size t times the speed of the viewer. If we denote the speed of the viewer by the vector v, we find:

$$\Delta E_{equi-latency} = vt \tag{5.28}$$

Assuming a uniform distribution of viewer motions v, we use the 1/12 just as in (5.23) and define:

$$\sigma_{\Delta E;equi-latency} = \sqrt{\frac{1}{12}} |v| t \tag{5.29}$$

This enables us to use the equivalent errors in (5.27) properly. We then obtain a bound on t and v:

$$|v| t < \frac{\alpha_{eye} \left| H^{z_H} \right|^2}{\left| P^{z_H} \right|_{\max}} \tag{5.30}$$

In [Pasm99] it was indicated that the latency should not exceed about 40 msec for head mounted display (HMD) systems. Equation (5.30) shows that the latency effects become less visible when the head-display distance H^{z_H} rises and the display-object distance P^{z_H} decreases. Since this holds clearly for our system compared to the HMD system, we assume that in our application larger latencies are allowed. This conclusion must be drawn with care, however, as (5.30) may not hold for HMD systems in which the display is attached to the viewer.

For a viewer about 1 m from the display and a scene with a size of about 15 cm, we find that $|v|t < 4$ mm (in approximation). With a system latency of $t = 40$ msec, the maximum speed at which the viewer may move his head is $|v|_{\max} \approx 10$ cm/s according to (5.30). This speed is in the order of normal head movements, but certainly people are able to move faster than this. This suggests that considerable effort has to be put in low latency rendering techniques for adaptive multi-viewpoint systems, as was done earlier for HMD systems [Pasm99].

5.4 Experiments

The goal of our experiments is to test the validity and applicability of the general visualization algorithm defined in section 5.2, and the bounds derived in section 5.3 for eye-tracking and system latency. For this we have performed a subjective test with nine different persons. Next we will discuss our test environment, followed by the subjective tests for the generic visualization algorithm and the eye-tracking and latency errors.

5.4.1 Test environment

Figure 5.9 shows our test environment. We implemented the general visualization algorithm in OpenGL, running on a Silicon Graphics Octane computer. We used a synthetic scene consisting of a cube with dimensions 12x12x12 cm, with a planar background about 1.5 cm behind the cube. The scene was positioned such that the cube was centered in the display. Both cube and plane were textured with a colored checkerboard pattern. Stereoscopic images were generated and shown on a 21” display with a (stereo) frame rate of 60 Hz and a resolution of 1280x512 pixels (the monitor was set in the so-called stereo mode that reduces the vertical resolution by a factor of two). The actual frame rate of synthesized images was about 30 frames/sec, thus below the display frame rate. LCD shutter glasses [Ster] were used to show the left and right images to the left and right eye of the viewer, respectively. The glasses were synchronized with the display via a wireless infrared system.

The eye tracking was done with a commercially available DynaSight 3-D sensor [Orig], which has an absolute accuracy of 2/2/8 mm in the $x/y/z$ directions and provides about 30 measurements per second. The tracker measures the head position H, defined exactly in between the eyes. The eye positions were derived by adding or subtracting half the inter-eye distance d_{eye} to or from the x coordinate found by the tracker. This assumes that the viewer does not rotate his head along the y and z axes. We used a fixed inter-eye distance of 6.5 cm. The tracker device was mounted on top of the display and centered manually. The measurements were transformed from tracker coordinates to display coordinates, involving the rotation around the x axis and vertical translation between display and tracker, see Figure 5.9. These parameters were measured by hand. Any additional tracking errors due to possible misalignment of the tracker with the display were not taken into account.

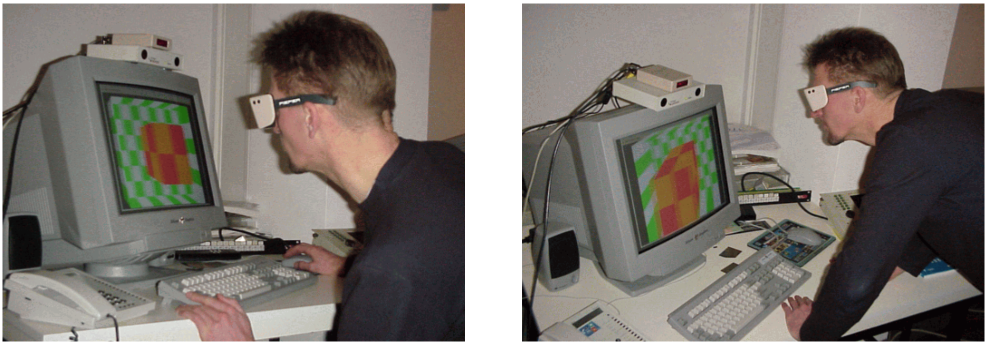
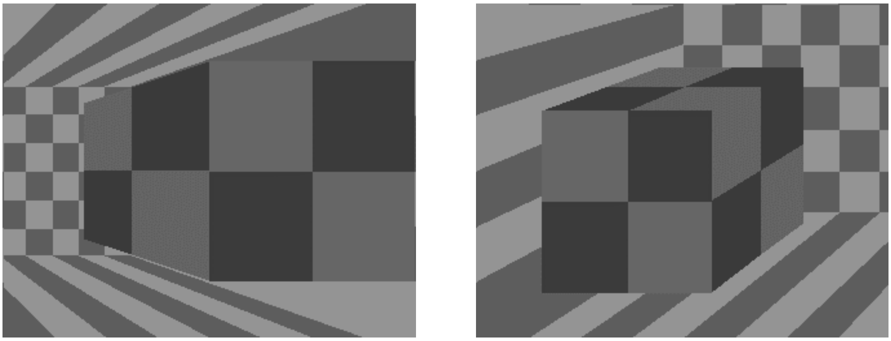


Figure 5.9 The test environment. The head tracker is on top of the display, the infrared transmitter for the LCD shutter glasses is on top of the tracker.

Before the subjective tests, the system performance was verified by a measurement of the geometry of the cube. The measurement was done in the way normal objects are measured, by using the system and holding a normal ruler beside the virtual cube. The size of the cube was found to be 11.8x12.7x12.6 cm. The manual measurement was only possibly with close viewpoints of about 50 cm from the display, due to the length of the human arms. The size of the cube was found to be invariant for all these close viewpoints in a range of about 90° around the display. The measurement in the depth direction was made possible by temporally repositioning the scene completely in front of the display, as discussed in section 5.2.2. This enables us to place the ruler next to the side of the cube without physically interfering with the display. The slight differences in size from the expected 12x12x12 cm can be due to the following. The pixel aspect ratio of the display in stereo mode was not exactly known, but approximated by two. Further, although no random eye tracker errors were noted, the tracker or the display-tracker calibration might introduce systematic tracking errors. Finally the thickness of the CRT display was about 0.5-1 cm, causing light rays to refract slightly from their intended paths.



a) -5, 0, 5

b) 10, 5, 15

Figure 5.10 Synthesized images for a viewer reading this thesis, with *x*, *y*, *z* positions given in cm relative to the image center.

Two left images of the generated stereo pairs are shown in Figure 5.10. When the reader takes a close look upon these figures with one eye closed and the other eye at the given coordinates, he observes the scene with correct geometry. The viewpoint coordinates for correct scene visualization have been adapted to the printing scale in this thesis. Even though the figures are monoscopic and do not provide viewpoint adaptivity, the square geometry of the cube can be observed very well. The effect of eye-tracking errors can be observed by moving slightly, causing the actual viewing coordinates to differ from the assumed coordinates.

5.4.2 Evaluation of the visualization algorithm

We evaluated the visualization algorithm by a subjective test with nine people with normal or corrected-to-normal (glasses) vision. All of them were engineers with considerable knowledge about the system. Each experiment took about 20-25 minutes. At the start, each viewer sat 1 m behind the display and confirmed that he could observe the stereo and motion parallax cues. We first asked some questions about the qualitative aspects of the visualization. Then tests were performed to see which position of the scene they preferred (centered in the display or e.g. behind it as discussed in section 5.2.2), and whether the stereoscopic and viewpoint-adaptive visualization contributed to the viewer’s comfort and 3-D experience of the scene.

General viewing quality

Qualitatively, all viewers mentioned that the sensation was pleasant. Everyone could see and describe the 3-D content of the scene (a cube in front of a plane). Several viewers mentioned that the cube was a bit stretched in both the y and the z direction, which is confirmed by our manual measurement. All viewers immediately observed that the LCD shutter glasses suffer from some cross-talk between left and right views. All of the viewers experienced this as annoying, either right from start or later on during the experiment. Further, the synthesized images were not rendered in anti-aliasing mode to allow for-high speed rendering. Especially due to the low vertical image resolution, this led to some annoying jittering of texture edges on the checker board patterns. This was observed unanimously. Similarly, all viewers experienced the system latency when changing viewpoint fast. They said the effect resembled watching a cartoon, or being drunk. Although it made the scene less realistic, it was not found annoying.

Position of the scene

The viewers were provided with the possibility to change the depth position of the scene manually until maximum viewing comfort was established. All viewers preferred a scene that was just behind or centered in the display plane. If the scene was positioned outside (in front of) the display, an impressive but fatiguing 3-D effect was experienced. The scene could not be positioned very far outside the display, since then it would fall outside the visibility region (see Figure 1.9 in Chapter 1). If the scene was positioned about 25-50 cm behind the display, most viewers had difficulty keeping the scene in focus, and then could no longer observe the scene in stereo. This is a typical effect of the accommodation-convergence conflict as shown in Figure 1.6 in Chapter 1. In this case, the motion parallax provided by the system was experienced as unnatural actual motion of the scene; as if the scene were attached to the viewer. Several viewers noted that with the scene positioned far behind the display, the system latency was much more pronounced, making them a bit dizzy.

Stereo or mono visualization

The effect of the stereo cue on the observed scene was tested as follows. Setting $d_{eye} = 0$ immediately provides the viewer with monoscopic visualization while maintaining motion parallax. At first, all viewers preferred the more sensational stereoscopic view. However, everyone made clear that the monoscopic view was far less fatiguing, and preferred this mode for longer viewing periods. This can (at least partly) be due to the annoying cross-talk of the LCD shutter glasses. The 3D-ness of the scene was less in mono viewing mode, but not absent, since the motion parallax cue was still present. When positioned at 2 m from the display, all viewers rated the stereoscopic visualization as comfortable as the monoscopic visualization. All but one viewer still experienced the stereoscopic depth cue clearly from this distance.

Motion parallax

We tested the effect of the motion parallax cue, the feature that is introduced by multi-viewpoint systems, by switching it on and off in a number of ways. If switched completely off, the system reduces to a non-adaptive stereo system. All viewers strongly disliked this after having experienced the motion parallax cue for some time. The scene still contains some depth, but any movement results in large, observable geometric distortions, described by our viewers as “the scene is elastic”. It appears that viewers can see the distortions much better when they just have experienced the adaptive visualization that the multi-viewpoint system offers.

Next to turning off all adaptivity, we also turned on adaptivity only in *xy* directions, discarding the systems’ adaptivity in the *z* direction of viewer motion, and finally turning on adaptivity only in the *x* direction. In both the *xy* and *x* case, the non-adapted components of the viewer position were fixed to the last fully adapted measurement before turning it off. In the *xy* case, the viewers could notice deformations in the scene when moving back and forth. Despite of the absence of adaptivity to viewer movements in depth, a significant range of movement was possible before the deformations became annoying (about 25% of the viewer-display distance). When we further decreased the system’s performance to *x* adaptivity only, all viewers saw a minor decrease in motion parallax. When they stood up and sat down again, they described the deformations as quite severe, but sitting in a chair, the vertical viewer motion is usually very little and they noticed hardly any effect.

From this we may conclude that enhancing a stereo system by viewpoint adaptivity only in the *x* direction yields the largest increase in performance. Adding adaptivity to viewer motion in the *y* direction is useful when it is expected that the viewer will actually use this direction of freedom. Finally, adding the viewpoint adaptivity in the *z* direction results in a fully adaptive multi-viewpoint system. This outperforms the *xy* system only if the viewer moves his head back and forth more than about 25% of the viewer-display distance.

Geometrical correctness of the scene

To test the geometrical correctness of the scene, we let the viewers measure the size of the cube using a ruler as discussed in section 5.4.1. We also asked them to guess the distance between the cube and the background, without the aid of the ruler. Table 5.1 shows the results.

Viewer	Size x	Size y	Size z	Distance cube-background
1	11.8	12.5	12.7	1.5
2	11.5	12	10	0.5
3	11.5	12.5	12	1
4	11.8	12.7	12.6	3
5	11.5	12	12	1.5
6	11.8	12.6	12.8	1
7	11.9	12.7	13	2
8	11.8	12.5	12.5	1
9	11.7	12.5	12.8	1.5
μ	11.7	12.4	12.3	1.4
σ	0.2	0.3	0.9	0.7
target size	12	12	12	1.5
author calibration	11.8	12.7	12.6	-

Table 5.1 Scene geometry assessed by the viewers in cm. The cube size was measured with a ruler, while the distance between cube and background was guessed.

The viewers provided results in mm or in half centimeters. The mean values and standard deviations of the cube dimensions show that the geometry of the observed cube was consistent within the group of viewers up to a few mm. The results are close to the target values, apart from the systematic deviation also noted by our own calibration measurement in section 5.4.1.

5.4.3 Evaluation of eye-tracking and latency errors

We examined the effect of eye-tracking noise on the scene visualization by deliberately adding noise to the measured eye positions. The viewers were positioned at three different viewing distances of 0.5, 1 and 2 m. Uniformly distributed noise was added to the measured viewer position, independently for both eyes and for the x, y and z coordinates of the eyes. The noise interval was the same for all these six coordinates. The interval was set to zero at the start, but we slowly increased it until the viewer could just notice the effects. Table 5.2 shows the results.

It must be noted in this experiment that the absence of anti-aliasing in our rendering algorithm continuously produced jitter on the display. All viewers explained that it was quite hard to distinguish the (just noticeable) eye-tracker noise from this jitter. This may explain the large variances compared to the mean values over the group of viewers.

If (5.27) is rewritten as a bound for uniform intervals instead of standard deviations, we obtain:

$$|\Delta E| < \frac{\sqrt{2}\alpha_{eye}\left|H^{z_H}\right|^2}{\left|P^{z_H}\right|_{\max}} \tag{5.31}$$

With $\alpha_{eye} \approx 6x10^{-4}$ and $\left|P^{z_H}\right|_{\max} \approx 8$ cm, we find $|\Delta E|$ intervals of 2.5 mm, 10 mm and 40 mm. Apparently the viewers are a factor two to four more critical than we expected.

Viewer	Viewer at 0.5 m	Viewer at 1 m	Viewer at 2 m
1	1.2	2.7	3.1
2	4	3	2
3	1.8	3	14
4	3.2	4.2	27
5	0.5	0.8	6.3
6	1.4	2.6	6.0
7	0.7	2.3	12
8	0.5	9.8	2.5
9	1.8	2.5	9.0
μ	1.7	3.4	9.1
σ	1.2	2.5	7.9

Table 5.2 Noise on the eye-tracker measurements that produce just noticeable artifacts. The noise is uniformly distributed in the interval given in mm. The noise was added independently to the x, y and z positions of both eyes.

In section 5.3.2 it was found that the bound on the z component of the eye-tracking error was over-restrictive. We verified this by applying noise to the z component only. Table 5.3 shows the just noticeable interval of uniform noise applied isotropically to all three coordinates versus applied to only the z coordinate. The results differ significantly: by a factor of about five; smaller than the expected factor 30 via (5.26).

Viewer	Isotropic	z only
1	2.7	23
2	3	7.8
3	3	25
4	4.2	17
5	0.8	14
6	2.6	6.3
7	2.3	4.5
8	9.8	59
9	2.5	9.8
μ	3.4	18
σ	2.5	17

Table 5.3 Effects of isotropic noise (from Table 5.2) versus noise in only the z direction. The noise is in mm on the eye-tracker measurements that produce just noticeable artifacts. The viewing distance was always 1 m.

Further, in section 5.3.2 we used both uniform interval lengths and standard deviations σ, which were converted by the factor $\sqrt{12}$. Table 5.4 shows whether this conversion is also valid for visual artifacts. From the results we observe that uniform and Gaussian noise produce similar amounts of visible artifacts, determined only by σ. Thus the conversion is valid.

Viewer	σ uniform noise	σ Gaussian noise
1	0.8	1.3
2	0.9	1.2
3	0.9	0.5
4	1.2	1.8
5	0.2	0.2
6	0.8	1.9
7	0.7	0.8
8	2.8	2.7
9	0.7	1.3
μ	1.0	1.3
σ	0.7	0.8

Table 5.4 Effects of uniform and Gaussian noise, in mm on the eye-tracker measurements that produce just noticeable artifacts. The viewing distance was always 1 m. The σ of the uniform noise is derived from Table 5.2 by dividing by $\sqrt{12}$.

To verify the bound (5.30) on the system latency, the viewers were positioned at 0.5 and 2 m and asked to move their heads in circular and ping-pong patterns at such a pace that they could just notice any geometrical scene deformation. The movement speed was measured very roughly by hand. Table 5.5 shows the results.

Viewer	Viewer at 0.5 m	Viewer at 2 m
1	-	-
2	10 cm/s	1-2 m/s
3	10 cm/s	∞
4	-	1-2 m/s
5	3 cm/s	10 cm/s
6	20-30 cm/s	∞
7	10 cm/s	50 cm/s
8	10 cm/s	20 cm/s
9	15 cm/s	-

Table 5.5 Maximum head movement speed that still does not produce visible artifacts due to system latency. The speeds are measured very roughly by hand. A “-“ indicates that no valid measurement was found. An ∞ means that no latency was observed at all.

The viewers found this task quite difficult. Now and then they reported that they saw artifacts (scene elasticity or “cartoon” like motions) at all times, or none at all. None of the viewers experienced the artifacts as annoying.

For the given viewing distances and $\left|P^{z_H}\right|_{\max} \approx 8$ cm, and (5.30) then predicts that $|v|t$ should be smaller than about 2 mm and 3 cm for viewers at 0.5 and 2 m, respectively. The latency of our tracker and rendering algorithm was not exactly known, but approximated as follows. The measurement rate of the tracker is about 30 Hz, which gives a delay of about 33 msec. The rendering algorithm was capable of following the 60 Hz frame rate of the display, introducing 16 msec delay. In total, this gives a system latency of at least 50 msec. This results in head speed bounds |v| smaller than 4 and 60 cm/s respectively. By order of magnitude, Table 5.5 is consistent with this prediction.

5.5 Conclusions

We derived a general image synthesis algorithm for multi-viewpoint systems with geometrically correct 3-D scene visualization on stereo displays. The algorithm renders two multi-viewpoint images from a 3-D scene model. The images are updated continuously to the current viewpoint of the viewer. The algorithm allows for any viewpoint, in contrast to many current algorithms, which provide only intermediate views with respect to a certain stereo camera setup (used for acquisition of the scene model). We argued that, in addition to the look-around feature provided by the visualization algorithm, manual adjustment of the position, orientation and scale of the scene are useful features. They allow the viewer to (simulate a) walk around a scene, to increase the effective motion parallax and to minimize the accommodation-convergence conflict.

We analyzed the effect of eye-tracking errors on the visualization. Using the finite resolution of the display and the human eye, a simple bound was found on the eye-tracking error. Below this bound, visualization errors cannot be observed. Current commercial tracker devices meet this bound, but it must be observed that the translation and orientation of the tracker with respect to the display must be calibrated up to the same accuracy as the tracker errors.

Using the eye-tracker bound, we derived a simple bound for the system latency between eye-tracker measurement and the displaying of the synthesized images. The bound limits the product of the system latency and the allowed viewer mobility. For reasonable latencies of e.g. 40 msec, the allowed viewer mobility is in the order of 10 cm/sec. Although this speed is in the order of normal head movements, people may move faster than this now and then. Therefore, low latency rendering techniques such as used in Head Mounted Display virtual reality systems [Pasm99] might also prove very useful in adaptive multi-viewpoint systems.

A subjective test with nine persons was performed to validate the general visualization algorithm, the theoretical bounds on eye tracking errors and system latency. Most importantly, we asked viewers for their preferences about scene position and viewing mode (mono, stereo or viewpoint-adaptive). For the test, the visualization algorithm was implemented on a computer platform with real-time rendering capabilities. A synthetic scene was shown on a stereo PC display on the basis of LCD shutter glasses. Head tracking was performed with a commercial device.

The geometry of the visualized scene was found to be fully correct. The viewers were able to measure the geometry of a synthetic scene with a deviation of only 5% from the true values. This accuracy is remarkable since a normal ruler was used on a virtual object positioned in front of the display. The theoretical bound on eye-tracking errors was also correct. For this we deliberately added noise to the eye tracker data and measured which amount of noise the viewers could just notice. We similarly checked the bound on system latency, and found it to be correct too.

All viewers preferred the viewpoint-adaptive system to a non-adaptive stereoscopic system. Starting with the non-adaptive stereo system, adding adaptivity in the *x* direction only yields the largest increase in subjective system performance. Especially for seated viewers, the additional gain of adaptivity in the *y* direction is less. Finally the effects of adding adaptivity in the *z* direction was only noticeable when the viewer motion in the *z* direction was more than about 25% of the viewer-display distance.

Using a fully adaptive system (*x*, *y* and *z* adaptivity), all viewers evaluated stereo rendering as more spectacular than mono rendering. However, for viewpoints close to the display (~1m), viewers considered that the cross-talk between left and right images was very annoying and that it made watching fatiguing. Most viewers therefore indicated that for longer viewing periods they would prefer the monoscopic rendering mode. For further viewpoints (~2m), all viewers preferred the stereo mode. In this case, the left and right images were more similar and the cross-talk did not provide any annoying artifacts.

All viewers preferred a scene centered in the display or slightly behind it. At least two theoretical reasons may contribute to this. First, for these scene positions the accommodation-convergence conflict is (almost) minimized. Secondly, the effects of eye-tracking errors and system latency decrease with the distance between scene and display.

In general we may conclude that the introduction of viewpoint adaptivity in stereo display systems is currently not only possible, but even desired. There are several directions for future research to improve the performance of the system we discussed. The geometrical correctness of the visualized scene is still limited by the system latency, introducing “elastic” scenes when the viewer moves too quickly. The results from fast rendering techniques in Head-Mounted Display systems [Pasm99] may provide a solution for this. The scene geometry would also improve if accurate calibration of the eye tracker with respect to the display was done. This is an open issue at the moment. In terms of the display, several factors can be improved. First, by minimizing the thickness of the display glass between the light source (e.g. phosphors in a CRT) and the air, the effects of refraction between glass and air are minimized, which improves the scene geometry. Further, current displays still cannot provide the accommodation cue for the eye, resulting in the accommodation-convergence conflict. This is one of the reasons why viewers prefer scenes that are visualized in the vicinity of the display, instead of before or behind it. Finally, our test showed that it is most important to diminish the cross-talk between left and right views of a stereo display to enhance subjective system performance.

Chapter 6

PANORAMA real-time 3-D visual communication system

6.1 Introduction

In this chapter we will describe the 3-D visual communication system designed and built in the European PANORAMA project [Ohm98, Pano98a, Pano98b]. The PANORAMA system is a two-way communication system, consisting of two identical adaptive multi-viewpoint systems using a stereo camera and autostereoscopic display, as shown in Figure 1.11 in the introduction of this thesis. The target application for the system is 3-D videoconferencing, with scenes consisting of a single human being of whom only the head and shoulders are visible in front of a uniform (textureless) background.

The PANORAMA system was the first real-time multi-viewpoint communication system ever built. To realize this, several partners from universities and companies throughout Europe collaborated for three years: from September 1995 to October 1998. The universities are from Delft (Netherlands), Hannover (Germany), Milan (Italy), Patras and Thessaloniki (Greece). The companies are Atomic Energy Association (AEA) (United Kingdom), CCETT (France), Deutsche Telecom (Germany), Heinrich Hertz Institute (HHI) (Germany), Intracom (Greece), Siemens (Germany) and Thomson (France). Further the international medical consortium OP2000 [Jong99] participated by providing field trials in the medical area.

Figure 6.1 shows one channel of the system, which also incorporates stereo audio, audio/video encoding/decoding with commercial MPEG-2 codecs, disparity coding with new special hardware, multiplexing, ATM transmission and subsequent demultiplexing. The system further included two devices to ensure synchronization of e.g. left/right video streams and correspondence field data. For more information about all these additional features we refer to [Ohm98, Pano98a, Pano98b].

In this chapter, we will examine the system parts as indicated by Figure 6.1 and their integration into a 3-D videoconferencing system. The parts treated are the cameras, the correspondence estimator, the specific choice of scene model, the multi-viewpoint image generator on the basis of interpolation, the autostereoscopic display and the eye tracker. In the treatment, we will make use as much as possible of the theoretical framework of the previous chapters. It must be kept in mind though that not all of the results and algorithms could be included in the PANORAMA system. This is due to implementation feasibility reasons and the fact that the system was designed in 1995 for project continuity reasons.

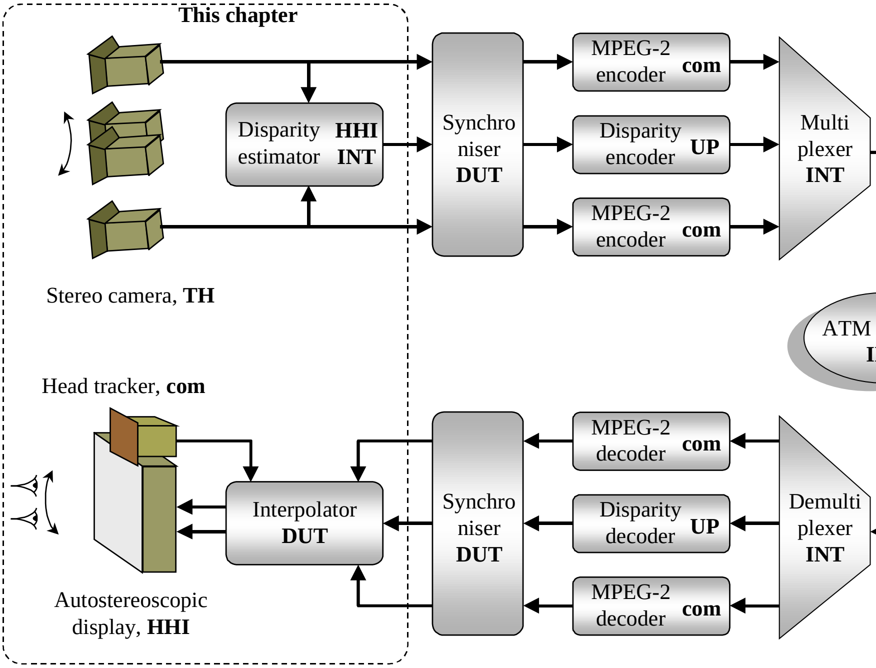


Figure 6.1 The PANORAMA adaptive multi-viewpoint system, build by the partners shown (com = commercial product, DUT = Delft University of Technology, HHI = Heinrich Hertz Institute, INT = Intracom, TH = Thomson, UP = University of Patras). The videoconferencing application uses two such systems, including an audio channel. This chapter treats the parts in the dotted area and their integration into a system.

The chapter is organized as follows. In sections 6.2 to 6.5 we will first describe a framework for a two-way multi-viewpoint system with an image-based scene model. In section 6.2 we review several scene models and define a specific image-based scene model that allows for feasible system implementations. In section 6.3 we show that the model can be acquired efficiently with a stereo camera using the results from chapters 2, 3 and 4. In section 6.4 we rewrite the visualization algorithm from Chapter 5 for the new scene model and derive several lower complexity algorithms. One of those consists of disparity compensated image interpolation such as employed by the PANORAMA system. It will be shown that under certain conditions geometrically correct scene visualization is possible with image interpolation. In section 6.5 we examine the specific constraints in a multi-viewpoint system due to our two-way videoconferencing application. Then, in section 6.6 we describe the specific details of the PANORAMA system, making full use of all previous results. In section 6.7 we report on the extensive subjective tests with the PANORAMA system, performed at Heinrich Hertz Institute by a professional marketing company. Finally, section 6.8 concludes the chapter.

6.2 Scene model for real-time systems

The scene model contains the photometric and geometric properties of the acquired scene. The choice of model influences both the analysis and synthesis parts of the system and thus the overall system complexity. In section 6.2.1 we will discuss several models and decide that image-based scene models are the most promising for a real-time system implementation. Then we discuss a specific image-based scene model in section 6.2.2 that is closely related to the PANORAMA scene model.

6.2.1 Different types of scene models

Many different kinds of models are available at the moment, such as wire frames, light fields, voxel maps, holograms and image based models. Wire frames are easily generated for synthetic scenes. However, the acquisition of real scenes into a wire frame is still an extremely demanding task that is not likely to be real-time feasible soon [Pano98a]. Light fields [Levo96] or ray spaces [Fuji96] describe all light rays that pass a certain surface that completely contains the scene. For each point on this surface (which has two positional coordinates), the intensity and color of a light ray in each direction (which has two directional coordinates) is described. Thus, a dense 4-D array is used, that contains an enormous amount of data. With only two cameras available, most of this space would remain undefined or contain redundant data. The same argument holds for holograms and voxel maps. Although the fringe patterns in a hologram are defined on a 2-D array, the resolution needed is extremely high. Voxel maps describe scenes very explicitly by dense 3-D arrays. Each entry either is empty, or contains some object point (or cube) with some emittance or reflectance properties. For medical applications with CT, PET and MRI scanners, this scene model is very natural. For scenes with opaque objects, normal cameras only record the surface of objects, and most of the voxel scene model would remain empty or undefined. Whenever computing power becomes available to allow for scene acquisition with more than two cameras, the camera-independent scene models such as wire frames, ray spaces, voxel maps and holograms may gain in attractivity.

In image-based scene models [Kang99], the photometry of the scene is stored in a normal camera image. Geometric information is stored in special types of images, such as depth maps [Tzov96], and correspondence fields that describe pixel correspondences in image pairs, see Chapter 4. The use of images as 3-D scene model is highly attractive from a complexity point of view, since the analysis and synthesis parts of the multi-viewpoint system then have images at both their inputs and outputs. At the same time, image based models are pixel-dense, that is, they contain a number of scene points in the order of 10^6, and thus they allow for high quality scene modeling.

6.2.2 Image-based scene model

In this section we will discuss an image-based scene model that we introduced in [Rede97b] and further investigated in [Rede00]. It is closely related to, but not exactly the same as the scene model used in the PANORAMA system, which we introduced and examined in

[Rede97a, Rede97c, Rede97d]. The model discussed here is more appropriate to describe the overall system mathematically. The PANORAMA scene model is functionally the same, but incorporates slight differences to enable efficiency both in terms of coding and hardware implementation. This will be dealt with in section 6.6.1.

The image-based scene model consists of two images $I_M(x_M,y_M)$, $D_M\,(x_M,y_M)$ and three scaling constants, K_x, K_y and K_z. Figure 6.2 shows an example of I_M and D_M. In this chapter, we will use the x_M and y_M coordinates centered in the images and pointing in the illustrated directions. The x_M, y_M run from – ½ N_x to ½ N_x and from – ½ N_y to ½ N_y respectively. This differs from the way images are normally indexed, but for clarity and simplicity in the equations we excluded this trivial transformation in this chapter.

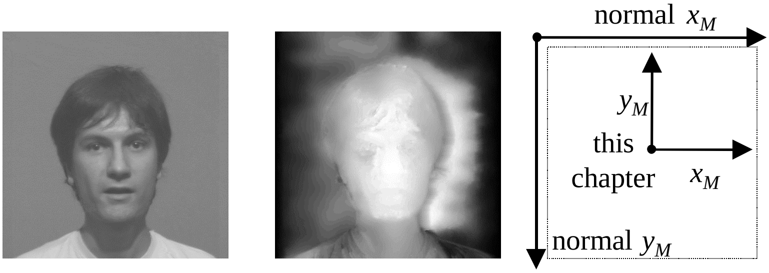


Figure 6.2 Luminance I_M, depth D_M and the image coordinates used in this chapter. The brightness in D_M is inversely proportional to depth.

For each x_M, y_M the model defines a scene point P with luminance $I_M(x_M,y_M)$ and 3-D position given by x_M, y_M, $D_M\,(x_M,y_M)$ and the three scaling constants according to:

$$\begin{bmatrix} P^x \\ P^y \\ P^z \end{bmatrix} = \frac{1}{D_M(x_M, y_M)} \begin{bmatrix} K_x x_M \\ K_y y_M \\ K_z \end{bmatrix} \tag{6.1}$$

The choice of (6.1) is guided by its compliance with the triangulation procedure for the D_M disparity field in (4.10), given that we use image coordinates as indicated by Figure 6.2. As can be seen, the z coordinate of P is inversely proportional to the 'depth' map D_M. Since luminance I_M is single-valued for each scene point, only Lambertian (diffuse emitting or reflecting) scenes can be modeled. Figure 6.3 shows the 3-D model that is represented by (6.1) and the $\{I_M, D_M\}$ pair of Figure 6.2.

This scene model has several advantages:

- For scene luminance, only a single image I_M is used which on its own provides compatibility with normal monoscopic video systems.
- For scene geometry, the depth map D_M can be converted in a simple way to 3-D coordinates via (6.1).
- The model contains only three additional parameters K_x , K_y and K_z, used for scaling.

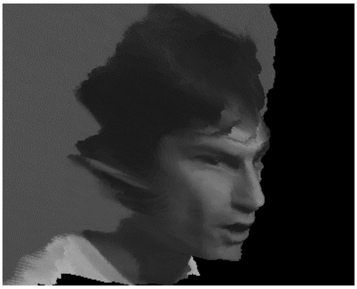


Figure 6.3 The 3-D model represented by $\{I_M, D_M\}$ of Figure 6.2.

In the next sections 6.3 and 6.4 we will see that this scene model can be acquired by a stereo camera using the techniques from chapters 2, 3 and 4, and that it integrates easily with the visualization algorithm from Chapter 5.

6.3 Acquisition with a stereo camera

Recently, hybrid cameras have become commercially available that can acquire an image and a depth map similar to D_M directly [Zcam]. At this moment, however, these cameras are still extremely expensive. Further, they employ active scene scanning, which puts limits on the scene dimensions.

The image-based scene model in Figure 6.2 can be obtained in a practical way by recording two images I_L, I_R with a stereo camera, after which the I_M and D_M images are constructed by image processing. We will discuss the stereo camera setup and calibration in section 6.3.1. In sections 6.3.2 and 6.3.3 the generation of the depth map D_M and the image I_M are treated. We will call I_M the center image since it is constructed by disparity compensated interpolation at center position.

The previously mentioned sections assume that scene points are visible in both the left and right images. In section 6.3.4 we will evaluate the effects of occlusions, that is, parts of the scene visible in only one image of the stereo pair.

6.3.1 Stereo camera setup and calibration

For the stereo camera we use the parallel setup, as outlined in section 2.4.9. We assume that camera calibration has been performed, see Chapters 2 and 3. Camera setups other than the parallel setup are evenly well possible, provided that the images from these cameras are rectified after calibration (section 2.4.9). Effectively this results in images from calibrated parallel cameras with parameters baseline b, focal length f and pixel size s_{cx}, s_{cy}. We use the additional c as subscript compared to the notation in section 2.4.9 since we will introduce a different pixel size for the display in section 6.4. The size of the images (camera CCDs) in pixels is N_{cx}, N_{cy}, assumed to be equal to the N_x and N_y of the scene model. Figure 6.4 shows an image pair I_L, I_R obtained with a stereo camera in parallel setup. This image pair was used to construct the I_M, D_M pair in Figure 6.2.

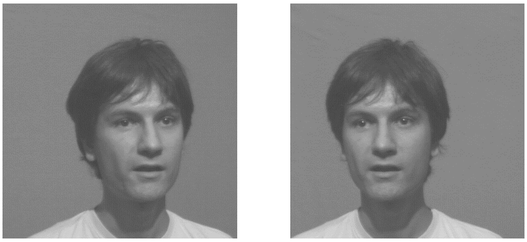


Figure 6.4 Original camera images I_L and I_R from a stereo camera in parallel setup.

6.3.2 Generation of depth map D_M and scaling constants

By disparity estimation, extensively discussed in Chapter 4, we construct the disparity field D_M, that is designed especially for the parallel camera setup. If we take the triangulation procedure (4.10) for this disparity field, use the image coordinates as in (6.1) and Figure 6.2, and use $B = ½\, b$, we find:

$$\begin{bmatrix} P^x \\ P^y \\ P^z \end{bmatrix} = \frac{B}{D_M(x_M, y_M)s_{cx}} \begin{bmatrix} x_M s_{cx} \\ y_M s_{cy} \\ -f \end{bmatrix} \tag{6.2}$$

This defines the three scaling constants in (6.1). Obviously, the four parameters B (half the baseline), f (focal length) and s_{cx}, s_{cy} (pixel size) contain one redundant parameter, as there are only three scaling constants in (6.1). This is conform to the horizontal pixel size reduction discussed in section 2.4.1. We will not use this reduction here for simplicity reasons. Later we will relate the three parameters to other parameters, and then all of them share the same unit of meters (opposed to the hpu unit that has to be used otherwise).

6.3.3 Generation of center image I_M

Each x_M, y_M is related to a scene point for which we find the 3-D position by (6.2). Now we must determine what is the luminance of this point. Via (4.7)-(4.9) we find:

$$\begin{aligned} x_L &= x_M + D_M(x_M, y_M) \\ x_R &= x_M - D_M(x_M, y_M) \\ y_L &= y_R = y_M \end{aligned} \tag{6.3}$$

We now have the coordinates of the scene point projections in the left and right images, I_L and I_R. We perform a weighted average of the luminances of the left and right image:

$$I_M(x_M, y_M) = (½ + ½\Delta)\cdot I_L(x_L, y_L) + (½ - ½\Delta)\cdot I_R(x_R, y_R) \tag{6.4}$$

Geometrically, this construction equals averaging the luminance of the left and right images via the correspondence vector as illustrated in Figure 4.21. The I_M image lies centered in the I_L and I_R images, as indicated mathematically by (4.7). The resultant I_M is the image as it would be obtained by a real camera positioned exactly in the center of the left and right cameras, hence the name center image.

Photometrically, we used the weight Δ, which we set to:

$$\Delta(x_M, y_M) = \frac{D_M(x_M + \tfrac{1}{2}h, y_M) - D_M(x_M - \tfrac{1}{2}h, y_M)}{h} \approx \frac{\partial D_M}{\partial x_M} \tag{6.5}$$

The Δ equals the derivative of the depth map D_M with respect to x_M, averaged over a horizontal path with length h. It is directly related to the orientation of objects in front of the stereo camera. Slanted objects appear with different sizes in the left and right images, see Figure 6.5.

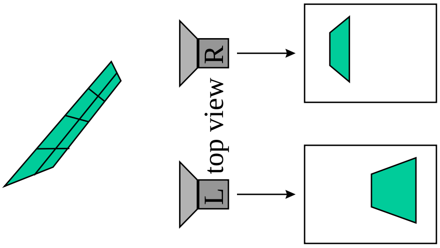


Figure 6.5 A slanted object appears with different sizes in the original left and right image. The object is best rendered in the center image using that original image that contains the object in highest detail.

For $\Delta = 1$, the apparent size of the object is zero in the right image. This particular object will be constructed in the center image solely with data from the left image. In the situation $\Delta = -1$ the role of left and right images are reversed. For planar objects, the apparent sizes are equal, which corresponds to $\Delta = 0$. Then the original left/right image data are averaged. This scheme ensures that each object in the scene will be rendered in the center image on the basis of the original image that contains the object with the highest resolution.

The h parameter governs the trade-off between adaptability to detail and sensitivity to noise in the estimated D_M. We found that the algorithm is quite insensitive to h, and that good results are found for $4 \leq h \leq 16$. Figure 6.2 shows the I_M image obtained with $h = 8$.

The weighting algorithm works only well when $|\Delta| \leq 1$, which can be ensured by the ordering constraint (4.11).

6.3.4 Occlusion effects

The scheme in sections 6.3.1 to 6.3.3 to acquire the I_M and D_M images on the basis of stereo imagery is defined for corresponding points in the left and right images. Whenever a part of the scene is visible in one image but occluded in the other image by some other part of the scene, no correspondence can be established. However, the acquisition scheme can still be used as follows.

The images in Figure 6.4 show a person of whom each ear is visible in only one image and occluded in the other. If the disparity estimator detects these objects as being maximally slanted ($|\Delta| = 1$) rather than occluded (Figure 6.6 shows an exaggerated example), then the objects will be represented in I_M and D_M as illustrated in Figure 6.2. Both ears are visible in the center image, where their horizontal size is compressed by a factor of two. Their geometry is represented by a continuous surface from the background to the foreground.

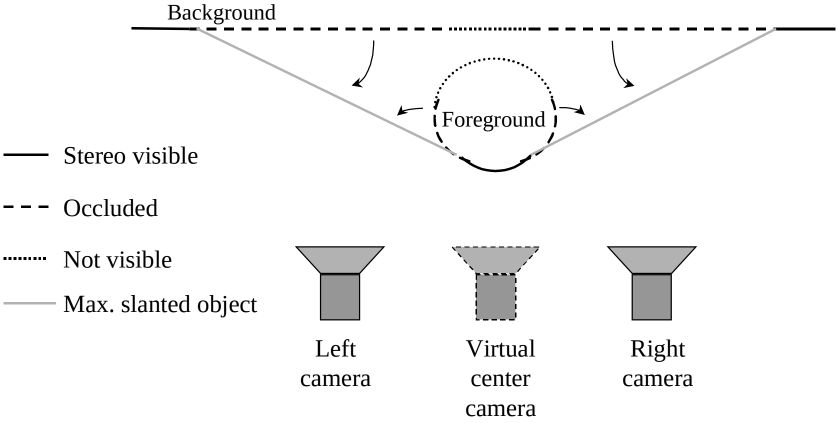


Figure 6.6 Occluded areas represented by maximally slanted objects. Depth cannot be measured for any area visible by only the left or right camera. It is interpolated between background and foreground parts that are stereo visible. The resultant virtual objects are maximally slanted objects. The virtual center camera sees all objects, with correct geometry at stereo visible areas and interpolated geometry otherwise. The arrows show the depth errors due to the interpolation (extreme example).

Of course, the depth assigned to the occluded areas is not correct. The arrows in Figure 6.6 indicate the errors made. Better depth measurements can be obtained if we use more complex estimation schemes, which e.g. extrapolate surface orientation around the occlusion, or use additional information from monoscopic features such as shape from shading, or use more cameras to acquire the occluded points in stereo with another camera.

Subjectively, our scheme provides a quite convincing center image (see Figure 6.2), including all occluded areas in both the left and the right image.

6.4 Scene visualization

In this section we will concentrate on the implementation of the generic algorithm for rendering multi-viewpoint images as discussed in Chapter 5, on the basis of the image-based scene model I_M, D_M. Head tracking and displays are commercially available, e.g. [Hein, Orig, Phil], and are not considered here.

In section 6.4.1 we will outline the generic algorithm. From that we will derive specific algorithms with significantly lower complexity in section 6.4.2. One of these algorithms is equivalent to the image interpolation which the PANORAMA system uses.

6.4.1 Generic visualization algorithm

According to section 5.2.2, we first have to define the translation, rotation and scale between the display and the scene model. Since our image-based scene model $\{I_M, D_M\}$ does not contain the rear side of the scene, rotating the scene manually for the 'walk around' possibility does not make much sense. We translate the scene by an amount T_z in the z_D direction towards the viewer, and then scale it by a factor S. As discussed in section 5.2.2, this provides a scene overview possibility and some means to minimize the accommodation-convergence conflict (see Figures 1.6 and 5.2). In combination with (6.2) we have:

$$\begin{bmatrix} P^{x_D} \\ P^{y_D} \\ P^{z_D} \end{bmatrix} = S\left(\frac{B}{D_M(x_M, y_M)s_{cx}} \begin{bmatrix} x_M s_{cx} \\ y_M s_{cy} \\ -f \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ T_z \end{bmatrix} \right) \tag{6.6}$$

Before we use this scene model in the generic visualization formulas (5.1) and (5.2) from Chapter 5, we will make some assumptions and introduce some variables.

We assume that the number of pixels is N_x, N_y for both the cameras and the display, and that their pixel aspect ratios are both equal to R:

$$\begin{gathered} \begin{bmatrix} N_x \\ N_y \end{bmatrix} = \begin{bmatrix} N_{cx} \\ N_{cy} \end{bmatrix} = \begin{bmatrix} N_{dx} \\ N_{dy} \end{bmatrix} \\ R = \frac{s_{cx}}{s_{cy}} = \frac{s_{dx}}{s_{dy}} \end{gathered} \tag{6.7}$$

The s_{dx} and s_{dy} are the display pixel size. Further, we introduce G, the scale difference between the camera and display pixels, and a system constant Z_{sys}:

$$\begin{gathered} G = \frac{s_{dx}}{s_{cx}} = \frac{s_{dy}}{s_{cy}} \\ Z_{sys} = Gf \end{gathered} \tag{6.8}$$

The f is the focal length (in meters) of the cameras. When we apply (6.6), (6.7) and (6.8) to the generic visualization formulas (5.1) and (5.2) we obtain (leaving out the transformation for the origin of the pixel coordinates):

$$\begin{bmatrix} Q^{x_I} \\ Q^{y_I} \end{bmatrix} = \frac{\frac{E^{z_D}}{Z_{sys}} \begin{bmatrix} x_M \\ y_M \end{bmatrix} + \left(\frac{B}{s_{dx}} - D_M(x_M, y_M) \frac{T_z}{Z_{sys}} \right) \frac{1}{B} \begin{bmatrix} E^{x_D} \\ E^{y_D} R \end{bmatrix}}{1 + \frac{D_M(x_M, y_M)}{S} \frac{s_{dx}}{B} \frac{(E^{z_D} - ST_z)}{Z_{sys}}} \tag{6.9}$$

Here, E is the eye location with its coordinates in the display reference frame as defined in Chapter 5, and Q is the projection on the multi-viewpoint image of a point P from the scene model $\{I_M, D_M\}$. With (6.9), we have the pixel coordinates x_I, y_I of Q, while the point P is given by x_M, y_M and $d_M = D_M(x_M, y_M)$. For the rendering process we refer to section 5.2.1.

Although (6.9) appears a bit complex, it describes everything that needs to be done for the synthesis of multi-viewpoint images in a fully viewpoint-adaptive system. As (6.9) only contains simple operations (additions, subtractions, multiplications and divisions) it seems possible to implement it in real-time.

The system constant Z_{sys} is related to the viewing angles of the cameras $[\Omega_{cx}, \Omega_{cy}]$ and the viewing angle of the viewer with respect to the display $[\Omega_{dx}, \Omega_{dy}]$, see Figure 6.7.

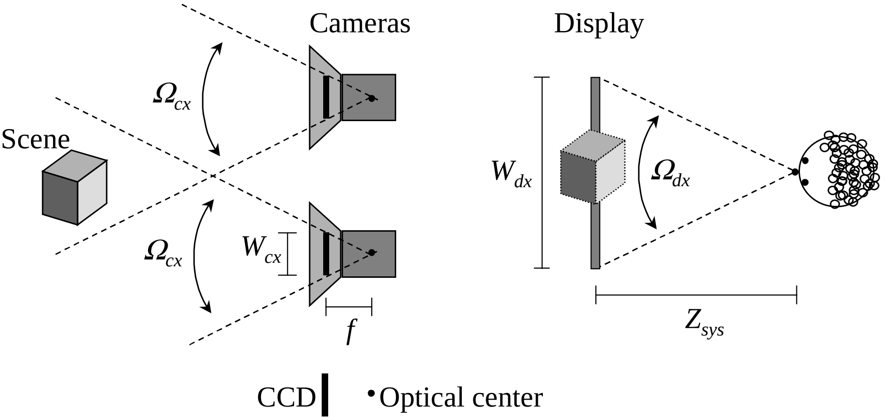


Figure 6.7 When the viewer is located at depth Z_{sys} in front of the display, his viewing angle to the display equals the viewing angle of the cameras. This figure is a top view showing only the Ω_x angles.

We find:

$$\begin{bmatrix} \tan\,\Omega_{cx} \\ \tan\,\Omega_{cy} \end{bmatrix} = \frac{1}{f} \begin{bmatrix} W_{cx} \\ W_{cy} \end{bmatrix} \qquad \begin{bmatrix} \tan\,\Omega_{dx} \\ \tan\,\Omega_{dy} \end{bmatrix} = \frac{1}{E^{z_D}} \begin{bmatrix} W_{dx} \\ W_{dy} \end{bmatrix} \tag{6.10}$$

The Ws are the sizes of the display and CCD chip in meters:

$$\begin{bmatrix} W_{cx} \\ W_{cy} \end{bmatrix} = \begin{bmatrix} N_x s_{cx} \\ N_y s_{cy} \end{bmatrix} \qquad \begin{bmatrix} W_{dx} \\ W_{dy} \end{bmatrix} = \begin{bmatrix} N_x s_{dx} \\ N_y s_{dy} \end{bmatrix} \tag{6.11}$$

Whenever the viewer is located a distance Z_{sys} in front of the display ($E^{z_D} = Z_{sys}$), the camera and display angles are the same.

6.4.2 Specific algorithms with complexity reduction

We will discuss five different algorithms based on (6.9) that may serve for real-time implementations with reduced complexity. The price to pay is a reduction in adaptivity to viewer motion (via E), and less freedom in the positioning and scaling of the scene (via T_z and S, respectively). One algorithm results in image-interpolation as used by the PANORAMA system.

Adaptivity in all directions, but restricted scene position and scale

The largest reduction of (6.14) is accomplished by discarding the disparity dependent division in the numerator. This can be done by setting $ST_z = E^{z_D}$, which results in:

$$\begin{bmatrix} Q^{x_I} \\ Q^{y_I} \end{bmatrix} = \frac{E^{z_D}}{Z_{sys}} \begin{bmatrix} x_M \\ y_M \end{bmatrix} + \left(\frac{B}{s_{dx}} - D_M(x_M, y_M) \frac{E^{z_D}}{SZ_{sys}} \right) \frac{1}{B} \begin{bmatrix} E^{x_D} \\ E^{y_D} R \end{bmatrix} \tag{6.12}$$

This means that the scene position and scale are coupled, and adapt with the viewers depth position. Since only the product of the two parameters is fixed, we can still set either T_z or S at will, e.g. set T_z to minimize the accommodation-convergence conflict or set $S = 1$ for correct scale. In the first case, the scale of the scene is coupled with the viewer's movements in the depth direction. In the second case, the scene makes 'counter' moves in the depth direction when the viewer moves. The scene geometry remains correct, but effectively the algorithm does not provide motion parallax in the depth direction.

Although a major complexity reduction is achieved, these effects are very unnatural making the reduction not appealing on its own.

Only adaptivity in horizontal and vertical directions

The 'counter' moves of the scene caused by (6.12) can be circumvented by setting $ST_z = Z_{sys}$ and requiring $E^{z_D} = Z_{sys}$. The $ST_z = Z_{sys}$ restriction stops the counter moves, since no coupling is present anymore with the viewer's position. By $E^{z_D} = Z_{sys}$, the adaptivity to viewer movement in the depth direction is lost and any movement results in geometric distortion of the scene. As shown in Chapter 5, these distortions can be neglected for reasonable deviations of E^{z_D} from the prescribed position. As shown in section 6.4.2, the viewing angles of the cameras and the viewer with respect to the display become equal.

Both restrictions together ensure $ST_z = E^{z_D}$ as used by (6.12). This results in:

$$\begin{bmatrix} Q^{x_I} \\ Q^{y_I} \end{bmatrix} = \begin{bmatrix} x_M \\ y_M \end{bmatrix} - \left(\frac{D_M(x_M, y_M)}{S} - D_{offset} \right) \frac{1}{B} \begin{bmatrix} E^{x_D} \\ E^{y_D} R \end{bmatrix} \tag{6.13}$$

with

$$D_{offset} = \frac{B}{s_{dx}} \tag{6.14}$$

In stereo systems, correct scene visualization can be obtained for a single fixed viewpoint (see Figure 1.5 in Chapter 1), provided that the left and right images are slightly shifted in the horizontal direction with respect to each other [Grin94, Konr99, Kutk94]. The offset (6.14) is equivalent to the shift when generalized to multi-viewpoint systems.

Figure 6.8 shows several images generated with (6.13) and $E^{x_D} \in \{-B, 0, B\}$, $E^{y_D} \in \{-½B, 0, ½B\}$ and $S = 1$. Subjectively, all generated images look very natural. Even though the original images were recorded by a horizontally displaced camera, the generation of vertically displaced viewpoints is easy and successful. The image in the center is I_M, and all the eight other images are extrapolated from I_M via (6.13). The images left and right next to I_M are reconstructions of the original left and right image (apart from the shift D_{offset}). The PSNR resemblance with the original images I_L and I_R is 43 dB. Clearly, as a side effect, the image-based model $\{I_M, D_M\}$ can encode the original stereo image pair very effectively.

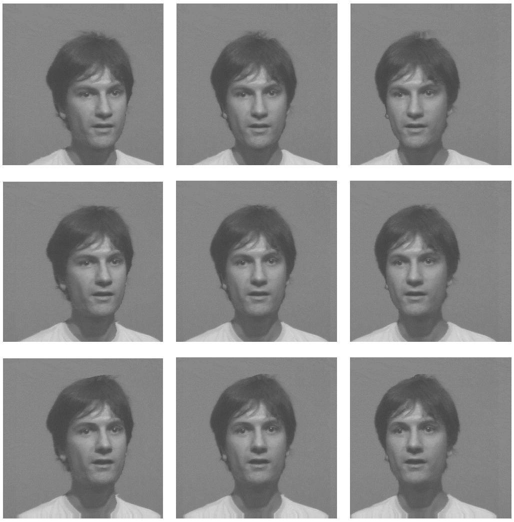


Figure 6.8 Synthesized images for several viewpoints.

Only adaptivity in horizontal direction

In Chapter 5 we saw that horizontal adaptivity is the most important factor for the introduction of motion parallax. If we discard adaptivity to vertical viewer movements by $E^{y_D} = 0$, and additionally set the scale constant $S = 1$, (6.13) becomes:

$$\begin{bmatrix} Q^{x_I} \\ Q^{y_I} \end{bmatrix} = \begin{bmatrix} x_M \\ y_M \end{bmatrix} - \left(D_M(x_M, y_M) - D_{offset}\right) \begin{bmatrix} E^{x_D}/B \\ 0 \end{bmatrix} \tag{6.15}$$

Due to fixing the scale to the correct $S = 1$, it automatically follows that $T_z = Z_{sys}$. We thus have no freedom anymore in setting the depth of the scene. Only scenes that are located at a distance Z_{sys} from the cameras that capture it, are visualized exactly centered in the display to minimize the accommodation-convergence conflict.

In terms of complexity, this restriction may be useful, since images are normally transmitted by line-wise scanning. Using (6.13), a full image needs to be stored by the algorithm, while (6.15) operates at a line-by-line basis.

An additional advantage of (6.15) is that it is valid without modification for image coordinates starting in the upper left corner as shown in Figure 6.2. Thus, the coordinate transformation need not be implemented at all. For (6.13) the same holds, apart from a sign change of R (it should be preceded by a minus sign to account for the y_I axis pointing downwards).

Restricted adaptivity in horizontal direction; image interpolation

For restricted viewpoints within $E^{x_D} \in [-B, B]$, the images returned by (6.15) are equal to horizontally shifted, disparity-compensated interpolations of the original left and right image (although generated here by extrapolation of I_M). The restricted viewing range corresponds exactly to the camera baseline $b = 2B$. In many algorithms for multi-viewpoint image interpolation, the shift D_{offset} needed for correct scene geometry visualization is not incorporated in (6.15) [Chup94, Liu95, Ohm97, Veig96]. The shift is incorporated in the PANORAMA system [Ohm98, Rede97f].

No adaptivity

The last reduction in complexity is achieved by discarding all adaptivity. Then, the images shown are always I_M for both eyes, resulting in a conventional, non-adaptive monoscopic system. In this case no processing is needed and also the D_M image need not be transmitted. Although it might seem easier to just record the I_M image by a camera, it is still worth the effort of performing the stereo acquisition to generate the I_M image.

In this way, a monoscopic videoconferencing application can be built in which direct eye contact is possible. A normal camera is always positioned on top of or next to the display, which inhibits direct eye contact. The two cameras of a stereo pair can be mounted left and right of the display, resulting in a virtual center camera mounted exactly in the display. A person can then be recorded straight in the face, while observing someone else on the display. A similar approach with a trinocular camera setup was adopted in [Liu95].

6.5 A two-way videoconferencing application

In this section we will discuss a symmetrical two-way multi-viewpoint videoconferencing system on the basis of image interpolation via (6.15) with $E^{x_D} \in [-B, B]$. Figure 6.9 shows one of the conferencing sites, where Alice communicates with (virtual) Bob. Here Alice is the viewer and Bob is the scene to be visualized. Bob's site is equal to that of Alice's (with the names exchanged).

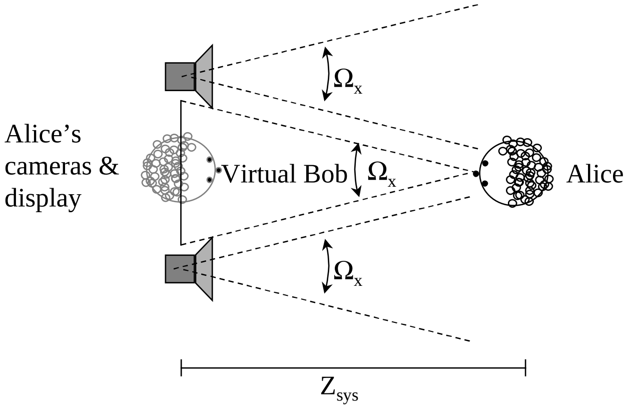


Figure 6.9 One site of a symmetric two-way videoconferencing application. For geometrically correct scene visualization, many constraints arise (parallel camera setup for acquisition algorithm, cameras and display at same depth for visualization algorithm, cameras left and right of the display for eye contact, image interpolation at the visualization side). If no special care is taken, the net result is that the viewing angles of Alice's cameras do not overlap at her.

In section 6.5.1 we will deal with restrictions that are specific for this application. These are due to the acquisition and visualization algorithms, the minimization of the accommodation-convergence conflict and the requirement of eye contact between Alice and Bob. In section 6.5.2 we will derive the net result, shown in Figure 6.9, that Alice's cameras cannot record her. Finally in section 6.5.3 several system solutions are proposed that solve this issue. At the same time, we explain how current interpolation algorithms may still produce correct scene visualization although the necessary shift D_{offset} is absent.

6.5.1 Restrictions due to system parts and the application

The requirements for the two-way videoconferencing application are described next. They are due to the acquisition algorithm, the visualization algorithm, the minimization of the accommodation-convergence conflict and the necessity of eye contact.

Acquisition algorithm

The acquisition scheme from section 6.3 requires a parallel camera setup.

Visualization algorithm

For correct scene visualization with image interpolation, the distance between a viewer and the display must be equal to Z_{sys} discussed in section 6.4.1. The algorithm (6.15) requires that the viewing angles of Bob's cameras and Alice's eyes to her display are the same Ω as discussed in sections 6.4.1 and 6.4.2. In a symmetrical system, the two one-way systems have the same Z_{sys} and Ω. Thus also Alice's cameras have viewing angle Ω, see Figure 6.9.

Minimization of accommodation-convergence conflict

To minimize the accommodation-convergence conflict, virtual Bob must be located in the center of Alice's display. For this, we found in section 6.4.2 that the real Bob must be located at a distance Z_{sys} behind his cameras. This is due to the fixed depth shift $T_z = Z_{sys}$ when using (6.15). Due to the two-way symmetry, this holds also for Alice. Since she is located at a distance Z_{sys} from her display, her cameras must be located at the same depth as her display, see Figure 6.9.

Eye contact

In videoconferencing, a vital feature is that the communicating persons have direct eye contact [Liu95]. Therefore, the heads of Alice and virtual Bob must be at the same height. If image generation is performed on the basis of interpolation, the left and right cameras recording Alice's head must be at the same height as the center of the display showing Bob's head.

6.5.2 Net effect of all restrictions

The restrictions due to minimizing the accommodation-convergence conflict and the necessity of the eye-contact require that the cameras are mounted left and right of the display. Then, the camera baseline b equals the display width W_{dx} plus some extra ΔW to account for the camera and display housing:

$$b = 2B = W_{dx} + \Delta W \tag{6.16}$$

Due to the restrictions from the acquisition and the visualization algorithm, the camera images do not overlap at the position of the Alice, see Figure 6.9. Subsequently, Alice cannot be captured in stereo and the system is useless.

6.5.3 Solutions for geometrically correct scene visualization

Several solutions are available to solve the issue of camera overlap found in section 6.5.2. We could relax the constraint that the Ωs be the same, but then we must accept that the system no longer provides geometrically correct scene visualizations. Further, we could place the cameras somewhere else, avoid the baseline constraint (6.16) and then project the cameras inside the display by means of a half-mirror, see Figure 6.10. This leads to large and cumbersome mechanical devices, however.

A solution that combines simplicity with correct scene geometry is to use cameras with shifted CCDs, as shown in Figure 6.11a. The shifts can be realized by physically repositioning of the CCDs, or by selecting a small portion of a larger CCD at the cost of resolution. The shifts can also be approximated by a camera setup that slightly converges, see Figure 6.11b. This is at the cost of some vertical disparity, in contrast with the real parallel camera setup that produces pure horizontal disparity. We can accounted for this by image rectification (see section 2.4.9), but for small convergence angles, the vertical disparity can be neglected [Rede97f].

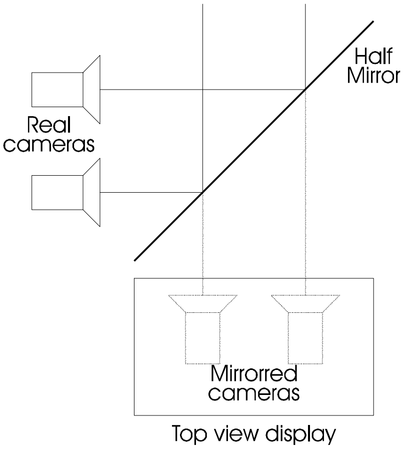
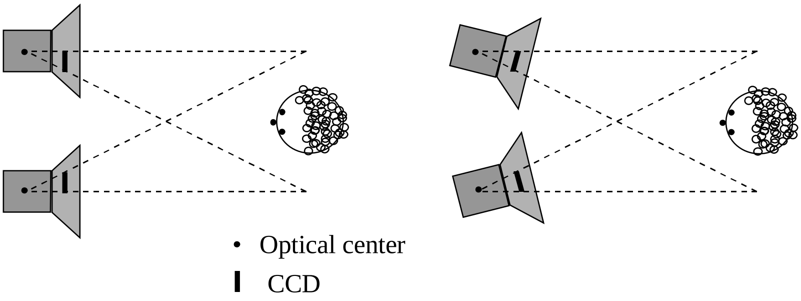


Figure 6.10 Avoiding the baseline constraint by the aid of a half-mirror.



(a) ***(b)***

Figure 6.11 Creating overlap by ***a)*** shifted CCDs or ***b)*** a converging stereo setup.

Shifting the left and right camera CCDs N_{shift} pixels to the right and to the left respectively (actually or effectively), lowers the x_L coordinate and rises the x_R coordinate of a corresponding pixel pair by N_{shift}. According to (6.6), this is equivalent with lowering D_M by N_{shift}, while x_M remains the same. Thus, in addition to increasing the camera overlap, we have implemented part of the shift (6.14). For correct scene visualization we must have:

$$D_{offset} + N_{shift} = \frac{B}{s_{dx}} \tag{6.17}$$

This offers a degree of freedom in the design of the system. However, the more the shift is assigned to the visualization side by D_{offset}, the less the camera images will overlap (the bundles in Figure 6.11 will overlap less). Moreover, if the converging setup of Figure 6.11b is used, the more the shift is assigned to the acquisition side by N_{shift}, the more vertical disparity will arise in the images, resulting in image distortion and small errors in the scene geometry. In such cases, a trade-off must be made between camera overlap and vertical disparity. A theoretical treatment of the trade-off can be found in [Rede97f].

Since

$$W_{dx} = N_x s_{dx} \tag{6.18}$$

we find with (6.16):

$$D_{offset} + N_{shift} = \frac{1}{2} N_x + \frac{\Delta W}{2 s_{dx}} \tag{6.19}$$

Figure 6.11 (both a and b) shows the ideal situation, where the shift is fully implemented at the acquisition side, with N_{shift} = ½ N_x and $\Delta W = 0$. This setup has at least two advantages. First, it enables multi-viewpoint interpolation algorithms with zero D_{offset} [Chup94, Liu95, Ohm97, Veig96] to provide correct scene visualizations. Secondly, it is easy to determine the scene dimensions. The width and height of the captured scene are exactly equal to the dimensions of the display, at a distance Z_{sys} from the cameras (see Figure 6.9). In front of the display, the scene has the shape of a screwdriver tip that ends at ½Z_{sys}. Behind the display, the scene continues infinitely.

Since the scene (a human head) is centered in the display as in Figure 6.9, it is clear that the display must at least have the size of a human head. In addition, if the person in the visualized scene moves his head, the display must be large enough to cover also this range of movement. For the person in the role of the viewer, the movement range is equal to the camera baseline as discussed in section 6.4.2. The movement ranges for a person in two-way communication, who is both viewer and scene, is equal to the camera baseline, since the display size and baseline are more or less the same via (6.16).

6.6 PANORAMA system

In this section we will discuss the details of the PANORAMA two-way multi-viewpoint videoconferencing system on the basis of image interpolation. In section 6.6.1 we deal with the PANORAMA scene model. This is effectively equal to the scene model introduced earlier in this chapter besides some implementational advantages. Then, we will deal with the acquisition and visualization algorithms in sections 6.6.2 and 6.6.3 respectively.

6.6.1 Image-based scene model

The $\{I_M,D_M\}$ introduced in section 6.2 allowed for a clear mathematical description in sections 6.3 to 6.5 for the system framework. The PANORAMA scene model consists of $\{I_L,I_R,S_M\}$, see Figure 6.12. Functionally, it is the same model, with some implementational advantages, and a slight image quality improvement capability. Photometric scene information is contained in I_L, I_R, the original left and right image. The scene geometry is contained in a special image, S_M, called the chain map, which is effectively equal to D_M [Rede97a, Rede97c, Rede97d].

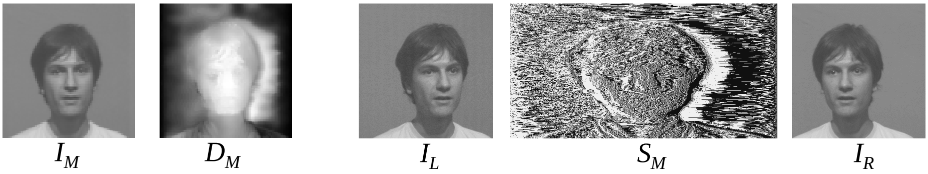


Figure 6.12 The $\{I_M,D_M\}$ scene model versus the PANORAMA scene model $\{I_L,I_R,S_M\}$.

We will now discuss the PANORAMA scene model images S_M, I_L, I_R, their relation with the $\{I_M,D_M\}$ scene model, and their format (resolution, color, etc).

Chain map image S_M

The disparity or depth map D_M is directly related to the chain map S_M by:

$$D_M(x_M, y_M) = \sum_{i=0,\frac{1}{2},1,1\frac{1}{2},\ldots}^{x_M-1} S_M(i, y_M)\frac{1}{2} \tag{6.20}$$

The D_M is the integral of S_M, and S_M is related to the derivative of D_M:

$$S_M(x_M, y_M) \approx \frac{\partial D_M(x_M, y_M)}{\partial x_M} \tag{6.21}$$

The S_M is defined as a binary valued function on an extended domain, that is, the integers together with the integers plus one half:

$$S_M(i, y_M) \in \{-1,1\} \qquad i \in \{0,\tfrac{1}{2},1,1\tfrac{1}{2},2,\ldots..\} \tag{6.22}$$

Figure 6.13 shows an example of D_M and S_M.

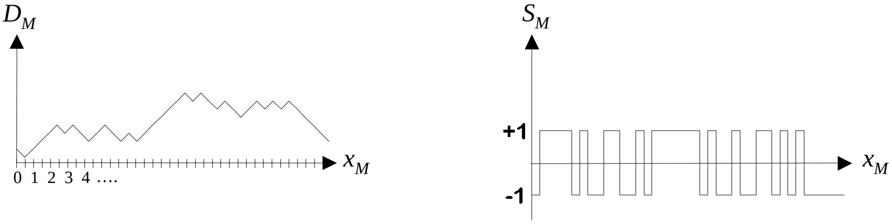


Figure 6.13 The depth map D_M versus the PANORAMA chain map S_M.

If we compare (6.21) and (6.22) with the ordering constraint (4.11), we see that the chain map inherently incorporates it. In [Rede97a, Rede97c, Rede97d] the chain map was introduced in a more complex form to model also occlusion areas explicitly, instead of via maximally slanted objects as in Figure 6.6. In those treatments, the chain map was subsequently reduced to the S_M discussed here. The use of S_M has several implementational advantages over D_M:

- There is no inherent upper or lower limit to disparity, opposed to e.g. an 8-bit D_M image.
- S_M is coding efficient, since it only requires $2N_x$ x N_y x 1 bit per image, which is 2 bit/pixel or equivalently 2 bit/scene point.
- S_M allows for efficient hardware implementations of the acquisition and visualization algorithms [Rede97d].

Luminance images I_L and I_R
The use of both left and right images in the PANORAMA scene model allows for the modeling of non-Lambertian reflecting surfaces (a scene point may have different luminance in the left and the right viewing direction). Also, the occlusion areas are represented with normal resolution, opposed to half resolution in the center image I_M. Finally, the system may render images at the most left and most right viewpoints with higher quality since these viewpoints correspond to the original images. These advantages are at the cost of more transmission bandwidth.

The three images I_L, I_R and S_M all have different horizontal coordinate axes. However, for each scene point indexed by x_M, y_M, the chain map S_M provides D_M given by (6.20). Then via (6.3) we can directly obtain the x_L and x_R coordinates and use the original images to obtain the scene point luminance.

Image format
The image format in the PANORAMA system is CCIR601, that is 720x576 pixels, 25 frames per second, interlaced, YUV color in 4:2:2 format. The chain map was vertically subsampled by a factor of four, yielding a vertical resolution of 144 lines per frame or 72 lines per field. This was due to the disparity estimator, which inherently provides this resolution. The chain map was linearly interpolated to full vertical resolution at the receiver side [Ohm98, Pano98a, Pano98b].

6.6.2 Scene acquisition

The scene acquisition process encompasses the definition of the position and size of the scene, the stereo camera and its setup, and finally correspondence estimation. We will elaborate on these topics next.

Scene position and size
For our application of videoconferencing, the scene consists of a human head and shoulders. This yields a scene with size of (about) 20x30x20 cm. For the position (see Figure 6.9) we select $Z_{sys} = 1$ m, which reflects a normal distance for both conversation and viewing a display.

Stereo camera and setup
We used a stereo camera in parallel setup. The cameras were mechanically very robust and precise, and contained high quality lenses. Therefore, the parallel setup could in principle be achieved without calibration or rectification. A large baseline of about 30 cm was used:

$$b = 30 \text{ cm} \qquad B = 15 \text{ cm} \tag{6.23}$$

This is the minimum baseline (6.16) due to the width of the display plus the housings of camera and display (see next section). The CCD chips in the cameras had size 1 cm by 0.75 cm. The number of pixels on the chip was 720x576. This yields the camera pixel size:

$$s_{cx} = 1.39\text{x}10^{-5} \text{ m} \qquad s_{cy} = 1.30\text{x}10^{-5} \text{ m} \tag{6.24}$$

The camera pixels have an aspect ratio R equal to 1.07. Substituting the display pixel size (6.29) discussed in the next section and (6.24) into (6.8), we find:

$$G \approx 32.00 \tag{6.25}$$

and then with $Z_{sys} = 1$ m we find via (6.8) we find the focal length f of the cameras:

$$f \approx 31.3 \text{ mm} \tag{6.26}$$

The overlap in the camera was implemented by a slightly converging camera setup as discussed in section 6.5.6. For the trade-off (6.19) of the shift at acquisition and visualization side, we derived theoretical solutions in [Rede97f]. However, especially in head-shoulder scenes with uniform background, the small scene geometry errors due to vertical disparity are hardly noticeable. Therefore we chose to maximize camera overlap as in Figure 6.11b. The shift was thus implemented completely at the acquisition side. With $B = 15$ cm and $Z_{sys} = 1$ m, the stereo convergence angle α as defined in section 2.6.1 equals about 17.3°, which gives negligible vertical disparity for our application [Rede97f].

With the choice of setup as in Figure 6.11b, the scene size is easily determined as discussed at the end of section 6.5.1. The scene width and height are those of the display, 21x29 cm, while the scene extends 50 cm out of the display in a triangular shape and infinitely far behind the display. This size is sufficient for the display of human heads that are slightly moving.

Correspondence estimation

Due to the (effectively) parallel setup, correspondence estimation is reduced to disparity estimation. This was performed by a hierarchical block matching algorithm, implemented in hardware. For each image field of 720x288 pixels the estimator yields a disparity field D_M with resolution 720x72, as discussed in section 6.5.2, which was then transformed into a chain map S_M automatically imposing the ordering constraint. The estimator had a pixel resolution and a maximum search range of 128 pixels with an adjustable offset. Via (2.65) we find for the depth accuracy at the scene center:

$$\left|\frac{\Delta z_M}{\Delta D_M}\right| \approx z^2 \frac{s_{cx}}{Bf} \approx 1\,\text{m}^2 \frac{1.4 \cdot 10^{-5}\text{ m}}{30\,\text{cm} \cdot 3.125\,\text{cm}} \approx 1.5\,\text{mm} \tag{6.27}$$

With $\Delta D_M = 1$ and a human head as scene we find that about 100 depth levels are needed, which is within the capability of the estimator. Details of the algorithm and its implementation can be found in [Ohm98].

6.6.3 Scene visualization

At the visualization side, we implemented a multi-viewpoint image interpolation algorithm in hardware, on the basis of (6.15) with $|E^{x_D}| \leq B$. The image shift (6.14) was almost completely implemented at the acquisition side, thus $D_{offset} \approx 0$ (zero, apart from slight manual adjustment to improve the subjective scene quality).

Since both the original left and right image are in the scene model, the luminance interpolation (6.4)-(6.5) was performed at the visualization side. The weighting incorporated an extra feature to enable exact reproduction of the original left and right image at the outer viewpoints $|E^{x_D}| = B$. The requirement for the weighting algorithm that the ordering constraint is fulfilled was automatically ensured by the chain map. The subsampled chain map from the acquisition stage was interpolated to full resolution at the visualization side. The chain map allows for efficient hardware implementations of (6.4), (6.5) and (6.15). For details of these algorithms and their hardware implementation we refer to [Rede97d].

A prototype autostereoscopic display was used with a viewpoint adaptive lenticular screen [Hein]. It was based on a 90° rotated VGA LCD. Its physical size was [Rede97f]:

$$W_{dx} = 21.3 \text{ cm} \qquad W_{dy} = 28.4 \text{ cm} \tag{6.28}$$

The number of pixels on the display was 480x640. This yields the display pixel size:

$$s_{dx} = s_{dy} = 4.44\text{x}10^{-4} \text{ m} \tag{6.29}$$

The display pixels are square and their aspect ratio R is 1. This yields a 7% difference between the aspect ratios of camera and display pixels, opposed to (6.7), which requires equal aspect ratios for correct scene visualization. This small difference was neglected.

The effective display resolution is 240x640, since the lenticular autostereoscopic display technique makes it necessary to multiplex the stereo image spatially over the screen. The generated multi-viewpoint images had CCIR601 format $N_x = 720$, $N_y = 576$ pixels, and thus were cropped horizontally (120 pixels at each side), extended vertically (32 black pixels at top and bottom) and subsampled horizontally by a factor two.

Due to the subsampled horizontal resolution, the horizontal size of the pixels is actually twice as large as in (6.29). Also for the pixel aspect ratio we find actually $R = 2$. However, since the resolution of both the display and the images is divided by two, the net effect is only slight blurring of the visualized scene, and (6.29) may still be used as effective pixel size for the determination of system constants as G, f and Z_{sys}.

Similarly, the number of pixels on the display differs substantially from that on the camera CCD chips, opposed to (6.7). This can be neglected, since the actual display is equivalent with a display with 720x576 pixels, part of which is covered due to the image cropping. As a positive side effect, this allows the cameras to be placed closer to each other than would have been possible with an actual 720x576 display. This is similar to the situation in Figure 6.10, where the display shown is the virtual 720x576 display, the dotted cameras are our actual cameras and our real display is located in between the cameras (not shown in the figure). This was however not done in the PANORAMA system, since it would also lead to a reduced horizontal movement freedom for the viewer.

A commercially available head tracker was used [Orig]. It measured the head position H as shown in Figure 5.5 (Chapter 5). Since (6.15) requires $E^{y_D} = 0$ and $E^{z_D} = Z_{sys}$, we defined the eye positions as:

$$E_L^{\sigma_D} = \begin{bmatrix} H^{x_D} - \frac{1}{2} d_{eye} \\ 0 \\ Z_{sys} \end{bmatrix} \qquad E_R^{\sigma_D} = \begin{bmatrix} H^{x_D} + \frac{1}{2} d_{eye} \\ 0 \\ Z_{sys} \end{bmatrix} \tag{6.30}$$

The inter-eye distance d_{eye} was set to 6.5 cm.

The tracker yields anistropic errors of 2, 2 and 8 mm in the x, y and z directions, respectively [Orig]. The resulting $\sigma_{\Delta E} = 2$ mm is just above the observation threshold of 1 mm, found in section 5.3.2 for a videoconferencing application (assuming a viewer-display distance of 1 m).

The implementation of the visualization algorithm could interpolate 256 different viewpoints between left and right images. With b equal to 30 cm, the viewpoint ‘density’ is about 1.2 mm. As discussed at the end of section 5.3.1, this is effectively equal to eye tracking errors of about 1 mm. As this is just at the observation threshold, the discretization is well chosen.

6.7 Experiments

The PANORAMA 3-D videoconferencing system was evaluated in a two-stage test by a professional marketing company, at the location of Heinrich Hertz Institute, Berlin. The internal PANORAMA document [Pano98c] reports on these and other tests. We will discuss the excerpt concerning our videoconferencing system.

In the first stage of the test, interviews with experts and a workshop with non-experts were held. The goal of this stage was to evaluate the main opinion about the idea, the advantages and the disadvantages of 3-D videoconferencing. The results were used to develop an optimal test strategy for the second stage. The second stage consisted of a subjective test with 16 persons, who evaluated the system under laboratory constraints.

In the sections 6.7.1 to 6.7.3 we will discuss the preceding interviews, the scenario for the subjective test and its results respectively.

6.7.1 Preceding interviews

Three experts were interviewed during one hour and a workshop with 18 non-experts was held. The topic was to evaluate the main opinions regarding the following questions:

- What are the strengths and weaknesses of the existing videoconferencing systems?
- What is expected of future and 3-D visualization?

It was found that the weaknesses of existing systems were the missing spatial feeling, the poor resolution, small images and a missing eye contact. However, a videoconferencing system was very economical (as compared to travelling), useful for exchanging information fast and directly, useful for intensive discussions between persons and groups, and finally a promising add-on in video communication.

A future videoconferencing system should be realized in high resolution, with a realistic view and flowing movement. The user should have the impression that he sits in the same room as the conference partner, i.e. experience telepresence. The non-experts said a 3-D feel is very important for an acceptable system. In contrast, the experts did not expect advantages from 3-D video and did not think that telepresence would play any role.

6.7.2 Scenario for the subjective test

From the first stage, it was clear that high resolution was generally thought to be necessary for future videoconferencing systems, while on 3-D and telepresence, no unanimous agreement was reached. The goal of the second stage is to verify these results with the PANORAMA system.

The subjective test consisted of 16 one-hour tests, with 16 persons using the real-time videoconferencing system. Figure 6.14 shows the system setup used for the tests. At the time of the test, a single hardware system was available. Therefore, our two-way system consisted of one 3-D link, used for the test, and a conventional monoscopic link.

In front of the test person, a conventional camera and a stereoscopic display were set up. Nearby, the interviewer of the marketing company and the operator were sitting. The second conferencing person sat behind a panel with a uniform background, with a stereo camera and a conventional display in front of him. An audio link improved the acoustic conditions, which lead to a more realistic scenario.

The influence of display quality was tested using two different displays: an autostereoscopic display with relatively low resolution (discussed in section 6.6.3), and a conventional PC monitor with high resolution, where stereoscopic images were shown with the test person wearing shutter glasses. This display is not useful in actual two-way 3-D videoconferencing systems; it was only applied to evaluate the high resolution. In our setup with one conventional link, the second person was not wearing special glasses so the test person could make unobstructed eye contact.

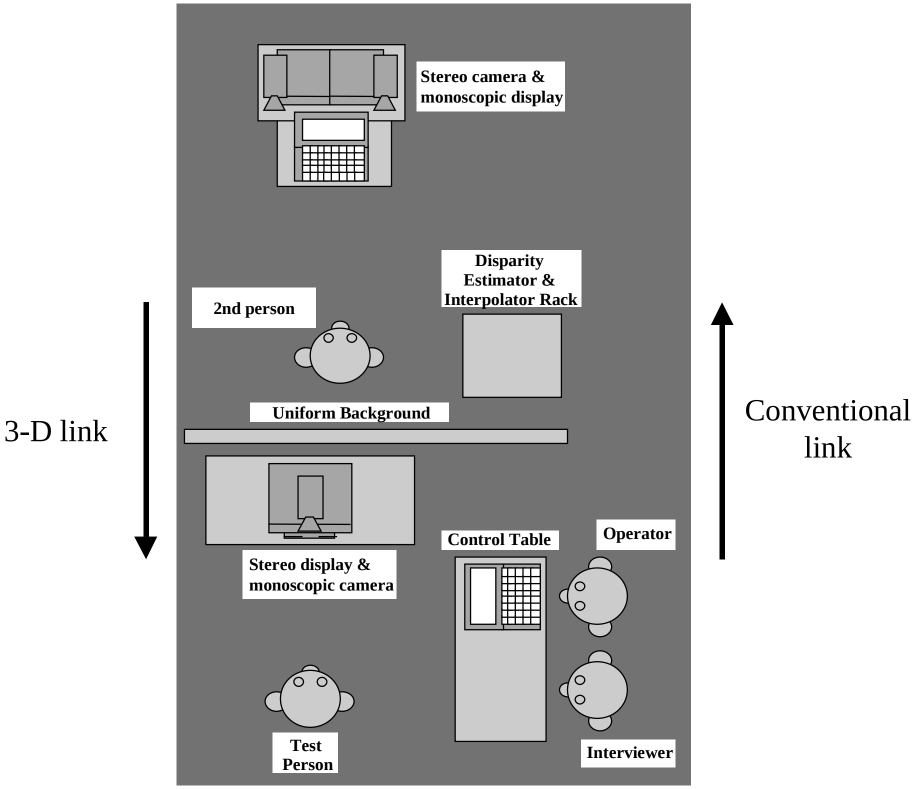

Stereo camera & monoscopic display
Disparity Estimator & Interpolator Rack
2nd person
3-D link
Conventional link
Uniform Background
Stereo display & monoscopic camera
Control Table
Operator
Test Person
Interviewer

Figure 6.14 The system setup for the subjective test.

The effect of 3-D and telepresence was tested by introducing three system modes: viewpoint-adaptive monoscopic mode, stereoscopic mode without viewpoint adaptivity and finally the full mode with viewpoint-adaptive stereoscopic visualization. Figure 6.15 shows some typical images generated by the PANORAMA system in the viewpoint-adaptive mode.

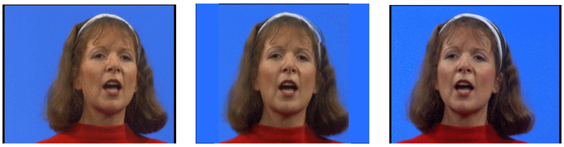


Figure 6.15 Viewpoint-dependent images generated by the real-time PANORAMA 3-D videoconferencing system.

For the system settings we used the results from section 6.6 as guidelines. The setup of cameras and display was performed manually until maximum viewing comfort was established. The deviations from the guidelines led to slightly incorrect geometry of the visualized person, but that was not noticed by any person during both the setup and the test itself (even when camera baselines up to 50 cm were used). The settings used were the same for all test persons during the entire test.

The two displays in the three modes result in six different situations. All of these were compared with a reference situation. For this, we used a conventional setup, namely monoscopic viewing without viewpoint adaptation at the PC monitor. In this situation, the PANORAMA system still provides direct eye contact as discussed at the end of section 6.4.2.

The following variables were evaluated with 1-2 questions in each test situation: image quality, spatial feeling, 3-D impression, naturality and eye contact. The test persons were to judge the visual impression without knowledge of the technical background and the applied mode. The rating scale was within the range 1 (worst) to 7 (best).

6.7.3 Results

It was found that the naturalness and eye contact was weighted similarly in all test situations. The ratings for image quality, spatial feeling and 3-D impression are given in Figure 6.16, Figure 6.17and Figure 6.18.

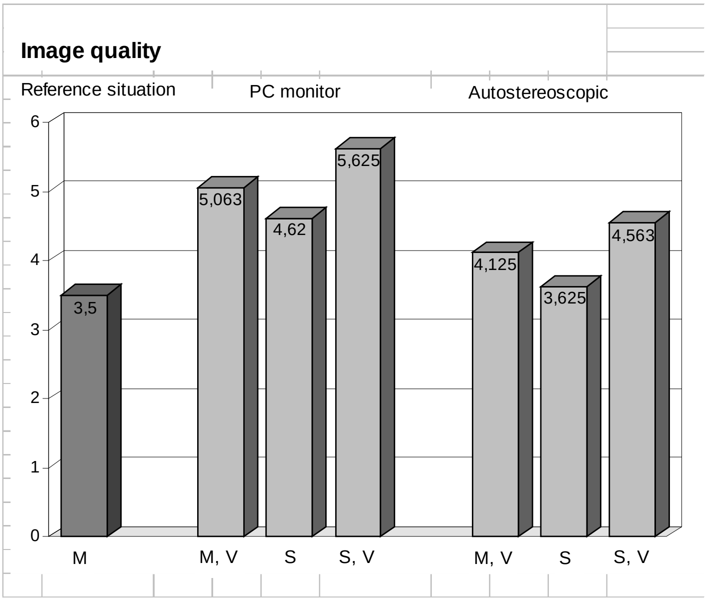


Figure 6.16 Subjective rating of image quality. M=monoscopic, S=stereoscopic, V=viewpoint adaptation, rating from 1 (worst) to 7 (best).

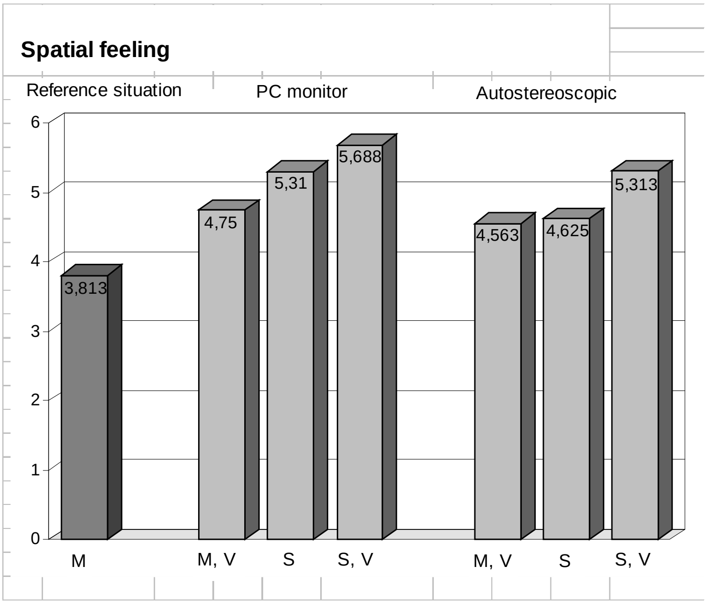


Figure 6.17 Subjective rating of spatial feeling. M=monoscopic, S=stereoscopic, V=viewpoint adaptation, rating from 1 (worst) to 7 (best).

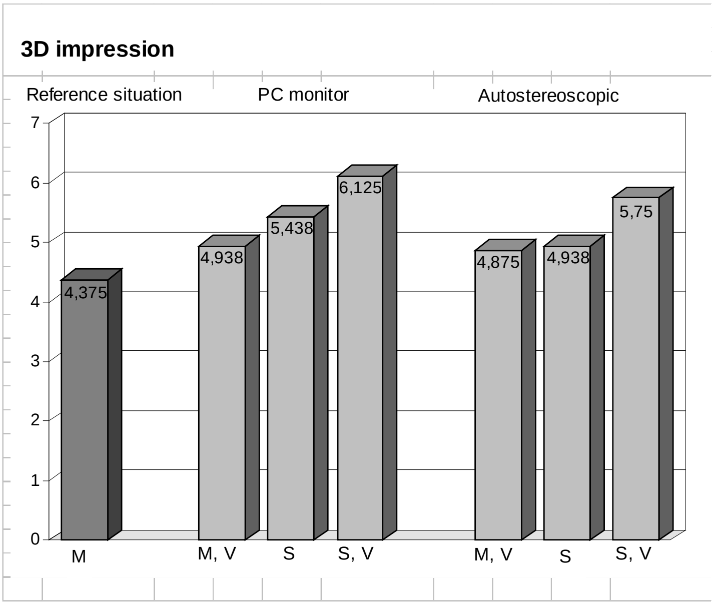


Figure 6.18 Subjective rating of 3-D impression. M=monoscopic, S=stereoscopic, V=viewpoint adaptation, rating from 1 (worst) to 7 (best).

The figures show that the higher resolution was preferred in all tests. Further, the image quality was rated best for the stereoscopic and for the viewpoint-adapted situation on each display. The same results were found for the spatial feeling and the 3-D impression. There was no difference between the results of experts and non-experts, regarding the 3-D impression. Experts were very impressed by the system and they modified their opinion of its merit.

6.8 Conclusions

In this chapter we described the PANORAMA 3-D visual communication system. It is an adaptive multi-viewpoint system as introduced in chapter 1, with scene acquisition on the basis of stereo camera and real-time disparity estimation and scene visualization with an image interpolation algorithm and an autostereoscopic display. The system has recently been built in hardware by a cooperation of several partners throughout Europe, including both universities and companies. It is the first system that actually realizes real-time, viewpoint-adaptive 3-D visual communication with natural scenes. The target application was 3-D videoconferencing, where two people communicate via a display and see each other in 3-D. Our main contributions discussed in the chapter are the introduction of a new scene model that forms the heart of the PANORAMA system, the acquisition of this model with a stereo camera, the transcription of the general visualization algorithm of Chapter 5 for this model and the derivation of overall system settings to obtain a correct 3-D impression.

Our new scene model is image based. It contains one image for scene luminance and one image for the scene geometry. As shown by the PANORAMA system, this scene model allows for real-time implementations of scene acquisition and visualization algorithms, and at the same time yields high-quality scene models (in the order of 10^6 scene points). The scene model provides backwards compatibility with conventional monoscopic television systems, it enables direct eye contact in monoscopic video-conferencing systems, and can be used to encode a stereoscopic image pair effectively.

We examined the acquisition and visualization algorithms on the basis of the new scene model. We showed that the scene model can be acquired with a stereo camera in parallel setup (or any other setup if the images are subsequently rectified), followed by disparity estimation and disparity-compensated interpolation of the stereo image pair. We combined the general visualization algorithm of Chapter 5 with the scene model of this chapter. Effectively, all images generated by this algorithm are extrapolations from the single luminance image in the scene model.

We derived several lower-complexity algorithms in order to obtain hardware feasibility. One of these algorithms equals image interpolation with respect to the original stereo image pair, as used in the PANORAMA system. We showed that geometrically correct scene visualization is still possible with such an algorithm, and we derived the conditions under which this is the case. Experiments were performed with typical video conferencing scenes and offline computer processing. These showed that high-quality, natural-looking images can be rendered for adaptive multi-viewpoint systems.

For the PANORAMA system, we were able to find settings that enable geometrically correct 3-D scene visualization. The feasibility of a real-time hardware implementation gives many constraints, which were all met. A constraint of more principal nature was identified for two-way videoconferencing applications: the recording cameras are attached left and right of a display, and thus the baseline is slightly larger than the display width. We showed that this poses a challenge, and that its solution is directly related to the conditions for correct scene visualization with image interpolation algorithms.

We reported on extensive subjective tests that have been conducted with the PANORAMA system. The main result of these tests is that the feeling of telepresence is greatly enhanced by the introduction of viewpoint adaptivity. The resulting motion parallax is a promising feature yielding a positive subjective impression. The system outperforms conventional monoscopic and stereoscopic systems. Even test persons with low expectations of 3-D techniques prior to our test changed their opinion in favor of 3-D after their experiences with the PANORAMA system. We may conclude that in the area of 3-D visual communication, multi-viewpoint systems are feasible candidates for real-time implementations.

Several issues are open for future research. For the acquisition part of our system, recently hybrid cameras have become commercially available [Zcam], which capture the image-based scene model directly, without processing. This may enhance the quality of the depth maps and at the same time reduce system complexity. At this moment however, these cameras are more expensive than the system built in the PANORAMA project. For the acquisition parts with normal cameras and processing, the more modern Markov Random Field (MRF) based correspondence estimators may be used, which have pixel resolution instead of the block resolution in our current approach. In Chapter 4 we showed that the computational requirements of MRF algorithms are not always exorbitant, which may lead to feasible real-time implementations. Further, multi-camera systems may be used to capture scenes with a more complex topology. This will require also more complex scene models, such as light fields or wire-frame models.

For the future of the visualization part in the system, the subjective performance of 3-D videoconferencing increases with the resolution of the display. Autostereoscopic displays must catch up with the resolution of e.g. current PC monitors. Further, the restriction of intermediate views can be relaxed. We showed that the general algorithm for any viewpoint has a complexity low enough to be implemented in real time, and leads to very natural multi-viewpoint images, even when the scene is acquired with only two horizontally displaced cameras. Further, we derived algorithms and system settings that ensured geometrically correct scene visualization, but at this moment it is not known to what extend a viewer actually needs that in a videoconferencing application. If this requirement can be relaxed, it may lead to more freedom in the camera and display setup. This was actually used during the tests with the PANORAMA system.

Chapter 7

Future outlook on 3-D visual communication systems

7.1 Current status of 3-D systems

In the area of visual communication, much research effort is currently being put in the enhancement of quality and telepresence (immersiveness) by 3-D visualization. The number of applications that benefit from such enhanced communication possibilities is obviously enormous. It enables improved videoconferencing in order to reduce business travelling, better remote assistance possibilities in the medical and industrial areas, new options for entertainment, advertising and educational purposes, and many more.

In the area of 3-D visual communication systems, one of the first systems was the stereoscopic system. Although the concept was already used in the 19th century for photography, it has not become used on world-wide scale in visual communication systems of today. The main reason for this is that the introduction of stereo was always accompanied by a loss in user-friendliness or in image quality. In cinemas, viewers had to wear special polarized glasses, while at home in front of the TV glasses were used that were colored red and green for the left and the right eye, respectively. The colors of the original TV program were lost, while the new colors caused head aches. The drawbacks of these stereo systems were caused by two facts. First, there were no so-called autostereoscopic displays available that worked without eye wear. Secondly, in the home television system the compatibility issue was of major importance. The TV displays had no polarization capability and were certainly not autostereoscopic. Further the transmission channels used PAL, NTSC, or other standards, leading to the principle of red/green color mixing.

In the past decades, major improvements have been made in technology for cameras, displays, transmission systems and signal processing. The introduction of digital technology enabled us to incorporate 3-D aspects far more easily than before, offering new options that sacrifice less image quality or user-friendliness. In recent years, many different systems have been developed. For example, a high-quality stereoscopic system was developed in the European research project DISTIMA. It was followed by the European PANORAMA project, in which a real-time autostereoscopic was built that, for the first time, incorporated another 3-D cue called motion-parallax. This cue enables the viewer to change his viewpoint on the visualized scene, e.g. to ‘look around’ objects. This was at the cost of providing imagery for a single viewer only. At MIT in the USA, prototype holographic systems have been built that allow for multiple viewers simultaneously. Holographic systems also show the 3rd and last visual 3-D cue: the eye lens accommodation cue that enables the eyes to focus on a particular object of interest. These holographic systems are, however, in a very premature state, requiring massive parallel computers and synthetic images, while still offering only very small display sizes.

In this thesis we investigated adaptive multi-viewpoint systems, which formed the basis of the PANORAMA system. These provide a viewer with stereoscopic images as well as viewpoint adaptivity (motion-parallax). In Chapter 6, we showed that these systems are the most promising candidates for real-time implementations at the moment and the near future: they use conventional cameras, (almost) conventional displays, conventional digital transmission channels and a substantial amount of state-of-the-art digital signal processing. At the same time, they enable 3-D scene capturing and visualization in full color and full resolution.

Subjective test with the PANORAMA system showed that the feeling of telepresence is greatly enhanced by the introduction of viewpoint adaptivity. The resulting motion parallax is a promising feature yielding a positive subjective impression. Even test persons with low expectations of 3-D techniques prior to our subjective tests, changed their opinion in favor of 3-D after their experience with the PANORAMA system. From this we conclude that 3-D aspects do and will contribute substantially in visual communication systems, since they provide the viewer with more realistic and natural impressions.

7.2 Directions for future research

Although significant advances have been made in the past few years in the area of 3-D visual communication systems, there are still many directions for future research. Systems on the basis of the multi-viewpoint concept such as the PANORAMA system, can be enhanced in many ways. In the scene acquisition part of the system, two cameras in a fixed setup were used to capture the scene. Ideally, their position, orientation and zoom may be changed during the recording (as is usual in monoscopic recordings). The real-time calibration of dynamic cameras is crucial and must be done by self-calibration. In chapters 2 and 3 we examined the most difficult case of self-calibration algorithms: using only two images of a scene from two cameras. We found a way to circumvent a theoretical proof that normally limits the use of such algorithms. Still, the reliability of our algorithm remains to be improved. Applying our algorithm with a multiple camera setup may be a solution. Further the speed of the algorithm must be increased several orders of magnitude.

Besides calibration, the acquisition of a scene in 3-D requires the estimation of corresponding pixels in different images. In Chapter 4, we have derived correspondence estimation algorithms for image pairs, especially suited for head-shoulder scenes in videoconferencing. We used Markov Random Field models with a special version of Simulated Annealing that, for the first time, combines high-quality with reasonable computation times. The robustness of this algorithm still needs to be improved, e.g. with a temporal consistency constraint. In terms of accuracy, a major area of research is the evaluation of our and similar algorithms with objective and meaningful quality measures. The measures we used were subjective, enabling rough evaluation but no quantitative comparison with other methods.

Further, multiple camera systems, possibly aided by other modality devices such as range cameras, allow for the capturing of more complex scenes. The analysis of these images requires improved image analysis techniques, e.g. better correspondence estimation algorithms that can deal with more than just head-shoulder scenes. We also need to use appropriate scene models, e.g. on the basis of light fields or wire frames. If all the aforementioned issues are solved, it leads to reliable, flexible and fully automatic methods for the real-time capturing of complex 3-D scenes.

For the 3-D scene visualization part of the system, The PANORAMA system used an image interpolation algorithm. In Chapters 5 and 6 we showed that this is sufficient for geometrically correct scene visualizations, whenever the viewer remains at a certain depth and height with respect to the display. We derived the algorithm for full viewpoint freedom and showed that it is possible to implement it also for real-time applications. Further, it remains to be investigated to what extent the geometrical correctness of the visualized scene is actually necessary. The more the requirement can be relaxed, the more freedom we have in the setup of cameras and displays.

The development of displays still has many possible directions. The telepresence feeling may be enhanced by higher resolution (shown in Chapter 6) and larger displays (e.g. room size). In Chapter 5 we showed with a subjective test that a major increase in viewer comfort can be established by a stereo display with better separation between left and right views. Further, the eye lens accommodation cue still remains to be introduced. Also, the number of people that simultaneously can see the scene undistorted is still limited. This may be overcome by e.g. fixed multi-view displays that show so many views simultaneously that a complete audience experiences the stereo and motion parallax cues without eye wear. Such displays are already being developed now for a moderate number of viewers (about five).

The investigation of transmission of 3-D scene models using conventional transmission channels is vital for the evolution of current monoscopic video systems towards 3-D systems on a global scale. For broadcast TV applications, this involves e.g. the Mpeg-2 compliant coding and multiplexing of the image-based scene model that we discussed in Chapter 6. Besides the TV application, other applications may come into play in this evolution scenario. Normal interpersonal communication is still mostly performed by phone. Ideally, the phone is replaced by the 3-D systems we examined, providing a wide-scale application. For actual videoconferencing, it is expected that multi-point communication is needed among much more than two people. Such applications are already emerging as research topics, e.g. the European VIRTUE project.

In the more distant future, where the transmission standards may be redefined completely, holographic systems may serve as the ultimate 3-D visual communication systems. Holographic displays serve all three 3-D cues (stereo, motion parallax and eye lens accommodation cues) for an unlimited number of people simultaneously. However, real-time acquisition of dynamical holograms will become possible only after technology has made a giant leap in terms of signal processing power and transmission bandwidth. In the meantime, the gap may be bridged by hybrid systems that acquire scenes with normal cameras, transmit them via conventional channels, but visualize them with holographic displays.

Appendix A

Geometry notation

A.1 Introduction

In this appendix we describe the notation used for geometry in this thesis. Sections A.2 to A.5 deal with general features that are used throughout the thesis. Then, only for camera calibration in Chapter 2, section A.6 deals with dot and cross products, and section A.7 deals with curved coordinates.

A.2 Reference frames, points and coordinates

Figure A.1 shows two right-handed Cartesian reference frames *A* and *B*, with origins O_A and O_B. Two points *P* and *Q* are shown and a vector $V_{P\ to\ Q}$ from *P* to *Q*. All points have coordinates in all reference frames:

$$\begin{array}{ll} P^{x_A} & x \text{ coordinate of point } P \text{ in reference frame } A \\ Q^{y_B} & y \text{ coordinate of point } Q \text{ in reference frame } B \end{array} \qquad \text{(A.1)}$$

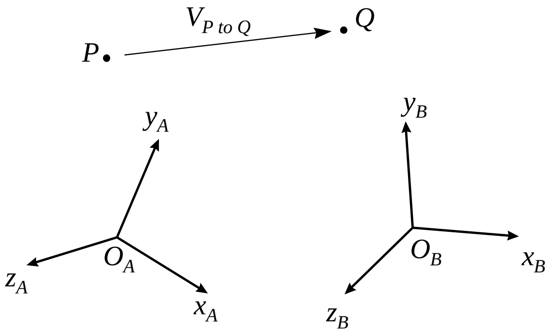


Figure A.1 Geometric notation.

We denote all three coordinates by an index σ in the range $\{x,y,z\}$. Then, the A coordinates of point P are denoted by:

$$P^{\sigma_A} \qquad \text{All three coordinates of point } P \text{ in reference frame } A \text{ indexed by } \sigma \tag{A.2}$$

To be able to write the coordinates in the usual notation, we use the following convention:

$$P^{\sigma_A} = \begin{bmatrix} -3 \\ 4 \\ 7 \end{bmatrix} \qquad \text{(the upper index } \sigma_A \text{ runs down)} \tag{A.3}$$

A.3 Vectors and components

A vector connects two points in space. It has no coordinates, but a difference in coordinates, which we call components:

$$V^{\sigma_A}_{P\,\mathrm{to}\,Q} = Q^{\sigma_A} - P^{\sigma_A} \qquad \text{components of vector } V_{P\,\mathrm{to}\,Q} \text{ in reference frame } A \tag{A.4}$$

From (A.4) it is clear that a number, associated to the component of a vector, invokes no less than three elements; A starting point, an endpoint and a reference frame. Only if all three of them are correctly defined, such a number is meaningful. It is the goal of our notation to make these elements very clear and to avoid mistakes.

A.4 Base vectors and matrices

The base vector in the x_A direction is denoted by V_{x_A}. Similarly, we can denote the other two base vectors, and write their components in the B references frame as:

$$V^{\sigma_B}_{\sigma_A} \tag{A.5}$$

In (A.5), the upper and lower index both run over $\{x,y,z\}$ separately. The nine numbers constitute what we call a base matrix. It relates the scale, rotation and skew between the A and B reference frames. The matrix equals the partial derivatives of the coordinates of the two frames:

$$V^{x_B}_{y_A} = \frac{\partial x_B}{\partial y_A} \tag{A.6}$$

Switching the upper and lower index is equivalent with inversion of the base matrix.

The components of an arbitrary vector V may be different from one frame to another, due to their different base vectors. The following relation holds between the components:

$$V^{\sigma_A} = V^{\sigma_A}_{\sigma_B} V^{\sigma_B} \tag{A.7}$$

On the right-hand side of (A.7), the double use of σ_B in a product, once as upper index and once as lower index, implicitly means summation over x_B, y_B and z_B. This is called the Einstein summation convention. Each index may appear at most once as lower index and once as upper index in a product. In this way, no matrix can be inverted by accident, since in (A.7) this would immediately give a conflict with the index of the vector.

The entries of the base matrix are denoted conventionally as follows:

$$V^{\sigma_A}_{\sigma_B} = \begin{array}{cc} \begin{bmatrix} \pi & -\sqrt{2} & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} & \downarrow \sigma_{\mathrm{A}} \\ \rightarrow \ \sigma_{\mathrm{B}} & \end{array} \tag{A.8}$$

Upper indices run downwards and lower indices run forwards. Two examples are:

$$V^{\sigma_A}_{\tau_A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \delta^{\sigma_A}_{\tau_A} \qquad\qquad V^{\sigma_A}_{\sigma_A} = 3 \tag{A.9}$$

Here we see how the τ index can be used to prevent summation. The δ is the Kronecker delta function:

$$\delta^{\sigma}_{\tau} = \begin{cases} 1 & \text{if the indices are the same} \\ 0 & \text{otherwise} \end{cases} \tag{A.10}$$

The nine numbers in a base matrix are completely determined by anisotropic scaling, rotation and skew. Scaling can be done simply by a diagonal matrix with non-unity entries:

$$V^{\sigma_A}_{\sigma_B\,;\,\mathrm{scale}} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \tag{A.11}$$

Negative values indicate mirrored coordinates. Rotation has three degrees of freedom by definition, but it influences all nine entries of the base matrix. We parameterize a rotation matrix by three Euler angles $\varphi_B^{A;\sigma}$:

$$V^{\sigma_A}_{\sigma_B\,;\,\text{rotation}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi_B^{A;x} & \sin\varphi_B^{A;x} \\ 0 & -\sin\varphi_B^{A;x} & \cos\varphi_B^{A;x} \end{bmatrix} \begin{bmatrix} \cos\varphi_B^{A;y} & 0 & -\sin\varphi_B^{A;y} \\ 0 & 1 & 0 \\ \sin\varphi_B^{A;y} & 0 & \cos\varphi_B^{A;y} \end{bmatrix} \begin{bmatrix} \cos\varphi_B^{A;z} & \sin\varphi_B^{A;z} & 0 \\ -\sin\varphi_B^{A;z} & \cos\varphi_B^{A;z} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.12}$$

A different type of parameterization is based on so-called quaternions [Azar95]. It avoids the computation of sines and cosines, at the cost of introducing complexity (the parameterization involves four real numbers and a constraint). We will use (A.12) since the parameters are directly related to physically measurable angles. If (A.12) is applied in (A.7), we see that rotation is performed first in the *z* direction (matrix on the right), then in the *y* direction and finally in the *x* direction. Figure A.2 illustrates the sign of the angles.

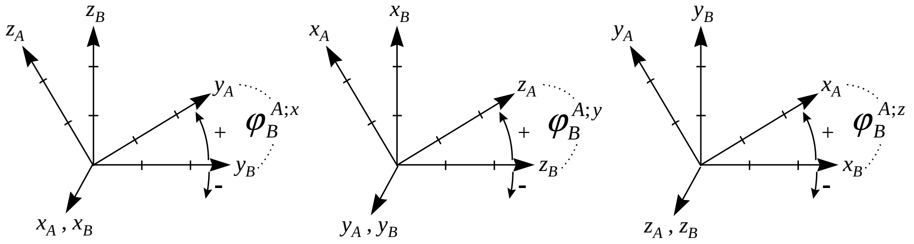


Figure A.2 Rotation between two reference frames.

Inversion of the rotation matrix (switching the indices) can be done by changing the sign of the angles and reversing the *xyz* order of the matrices in (A.12). Skew can be modeled by three numbers in a triangular base matrix (see Figure A.3):

$$V^{\sigma_A}_{\sigma_B\,;\,\text{skew}} = \begin{bmatrix} 1 & k_y^x & k_z^x \\ 0 & 1 & k_z^y \\ 0 & 0 & 1 \end{bmatrix} \tag{A.13}$$

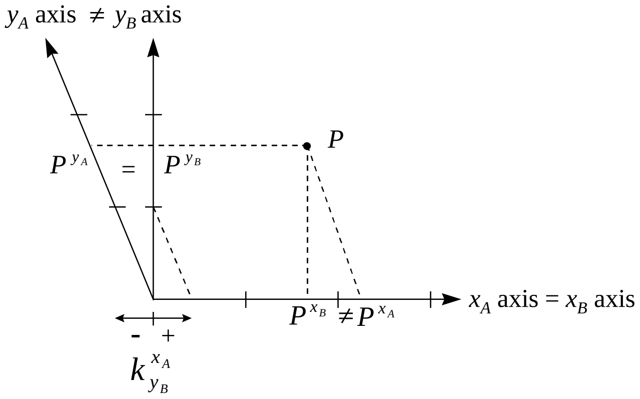


Figure A.3 Skew between two reference frames via an upper triangular base matrix. Note that the x_A and x_B coordinates are different and the y_A and y_B coordinates are the same, while the coordinate axes suggest the opposite.

In combination with scaling and rotation, other types of skew can be described, such as those given by a lower triangular matrix.

A.5 Transforms between reference frames

The coordinates of a point *P* in two reference frames *A* and *B* may differ due to differences in the position and orientation of the two frames. The relative position of the *A* and *B* frames can be denoted by two different vectors $V_{A\text{ to }B}$ and $V_{B\text{ to }A}$ going from one origin to the other, which have different components in the reference frames *A* and *B*. Thus, if no other reference frames are considered, the translation can be denoted in no less than four different ways. We will generally use the following notation that relates components to coordinates:

$$V_{A\text{ to }B}^{\sigma_A} = O_B^{\sigma_A} \tag{A.14}$$

Then, using the base matrix from the previous section, we can relate the *A* coordinates of *P* to its *B* coordinates.

$$\begin{aligned} P^{\sigma_A} &= V_{\sigma_B}^{\sigma_A}\left(P^{\sigma_B} - O_A^{\sigma_B}\right) \\ &= V_{\sigma_B}^{\sigma_A} P^{\sigma_B} + O_B^{\sigma_A} \end{aligned} \tag{A.15}$$

This can be seen from the last two terms of:

$$P^{\sigma_A} = V_{O_A\text{ to }P}^{\sigma_A} = V_{\sigma_B}^{\sigma_A} V_{O_A\text{ to }P}^{\sigma_B} = V_{\sigma_B}^{\sigma_A}\left(V_{O_A\text{ to }O_B}^{\sigma_B} + V_{O_B\text{ to }P}^{\sigma_B}\right) = V_{\sigma_B}^{\sigma_A}\left(V_{O_B\text{ to }P}^{\sigma_B} + V_{O_A\text{ to }O_B}^{\sigma_B}\right) = V_{\sigma_B}^{\sigma_A}\left(V_{O_B\text{ to }P}^{\sigma_B} - V_{O_B\text{ to }O_A}^{\sigma_B}\right) \tag{A.16}$$

If *O* and *V* are given with *A*/*B* indices in some up/down order, the other order can be computed, *V* via matrix inversion, and *O* by multiplying it by -*V*. The latter can be seen if one extracts the O_A term in (A.15) and requires identity.

A.6 Dot and cross products

The dot product $V_1 \cdot V_2$ of two vectors is defined separately for each reference frame. For frame *A* it is denoted by:

$$V_1 \cdot V_2 \ \text{ (in frame } A\text{)} = \delta_{\sigma_A,\tau_A} V_1^{\sigma_A} V_2^{\tau_A} \tag{A.17}$$

This yields a scalar. For $V_1 = V_2$, the dot product defines the square of length of the vector. The cross product is defined as:

$$V_1 \times V_2 \ \text{ (in frame } A\text{)} = \varepsilon_{\sigma_A,\tau_A}^{\mu_A} V_1^{\sigma_A} V_2^{\tau_A} \tag{A.18}$$

The ε is known as the completely asymmetrical Levi-Cevita tensor, which is defined by:

$$\varepsilon_{\sigma,\tau}^{\mu} = \begin{cases} 1 & \text{if } \sigma\tau\mu \text{ is an even permutation of } xyz \\ -1 & \text{if } \sigma\tau\mu \text{ is an odd permutation of } xyz \\ 0 & \text{otherwise (some indices are the same)} \end{cases} \tag{A.19}$$

The cross product yields a vector indexed by μ. If ε is applied to three vectors, it yields the triple product that defines volume of the parallellopipedum given by the three vectors, see Figure A.4.

The ε tensor always has a number of indices equal to the number of spatial dimensions under consideration. If applied to two vectors in a two-dimensional space, the cross product yields a scalar that defines area (see Figure A.4):

$$V_1 \times V_2 \ \text{ (in 2-D frame } A) = \varepsilon_{\sigma_A,\tau_A} V_1^{\sigma_A} V_2^{\tau_A} = V_1^{x_A} V_2^{y_A} - V_1^{y_A} V_2^{x_A} \tag{A.20}$$

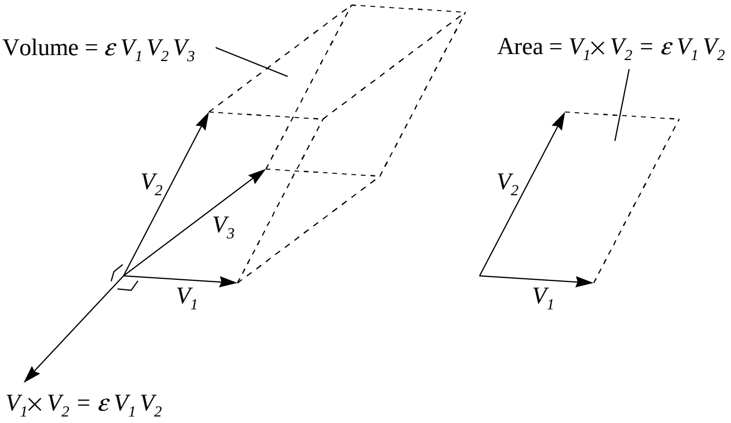

Volume = $\varepsilon V_1 V_2 V_3$
Area = $V_1 \times V_2 = \varepsilon V_1 V_2$
V_2
V_3
V_1
V_2
V_1
$V_1 \times V_2 = \varepsilon V_1 V_2$

Figure A.4 The cross product measures volumes in 3-D spaces and area in 2-D spaces.

A.7 Curved coordinates

If two reference frames A and B have a base matrix that is not constant throughout space, the coordinates of A and B are curved with respect to each other. Figure A.5 shows an example in two dimensions. The coordinates in this example are related by:

$$\begin{aligned} x_A &= x_B \\ y_A &= y_B + x_B^2 \end{aligned} \tag{A.21}$$

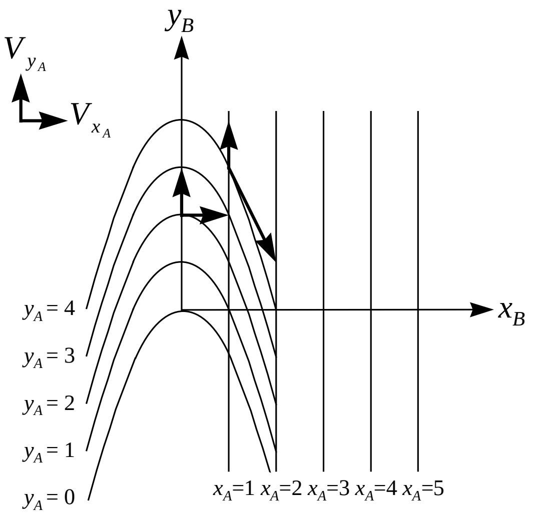


Figure A.5 Curvature between *A* and *B* frames.

This yields a base matrix in which one entry depends on position:

$$V_{x_B}^{y_A} = \frac{\partial\ y_A}{\partial x_B} = 2x_B \tag{A.22}$$

Curvature is characterized by a non zero curvature tensor $V_{\sigma_A,\tau_A}^{\sigma_B}$ (or $V_{\sigma_B,\tau_B}^{\sigma_A}$), which contains the partial second derivatives of the coordinates:

$$V_{x_B,x_B}^{y_A} = \frac{\partial^2 y_A}{\partial x_B^2} = 2 \tag{A.23}$$

Since the tensor is symmetrical in the lower indices, the number of independent parameters is 6.

Appendix B

Epipolar geometry

In this appendix we will describe epipolar geometry, which is a property specific to stereo cameras. Here we will treat the basics, which suffices for this thesis. For more detailed information about this topic, including extensions to three and more cameras, we refer to [Faug93, Truc98].

Figure B.1 shows a stereo (pinhole) camera. The optical centers (centers of the lenses of actual cameras) are O_{LFL} and O_{LFR}, the origins of the left and right lens reference frames. The line through these two optical centers is called the baseline. Any plane in 3-D space that contains the baseline is called an epipolar plane. All scene points in such a plane are projected on a line in each of the images. These lines are the epipolar lines. A pair of epipolar lines that share the same epipolar plane are called conjugate epipolar lines. If two points from the image pair correspond, they must lie on conjugate epipolar lines. This is called the epipolar constraint. It reduces the set of possible correspondence candidates for a point in the left image from all points in the right image to only those on the conjugated epipolar line in the right image.

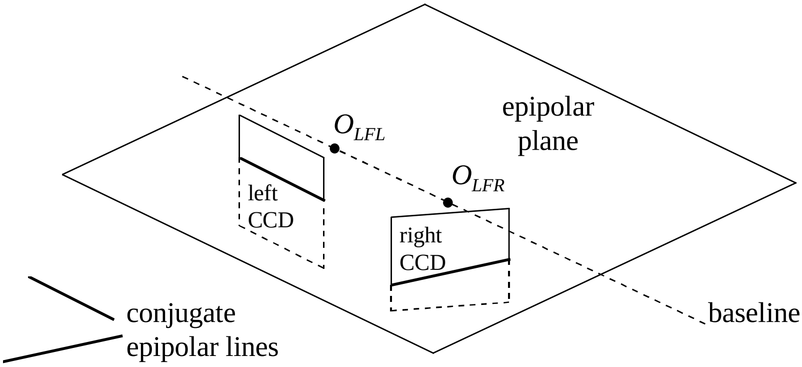


Figure B.1 Epipolar geometry.

Figure B.2 shows the position of the epipolar lines for parallel and convergent camera setups.

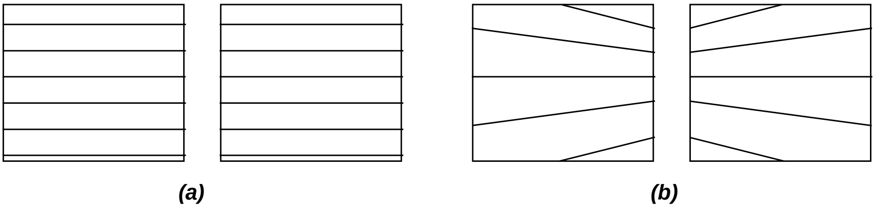


Figure B.2 Correspondences are constrained to conjugate epipolar lines, ***a)*** parallel camera setup, ***b)*** convergent camera setup.

For pinhole cameras, the epipolar lines are straight. Due to lens distortion [Weng92] the epipolar lines may become curved.

Appendix C

Gibbs and Markov random Fields

In this appendix we describe the basic characteristics of Gibbs and Markov Random Fields (GRFs and MRFs), sufficient for Chapter 4 of this thesis. For a thorough introduction to this subject we refer to [Gema84].

Markov Random Field (MRF) models can be used to model interactions between a large number of stochastic variables, arranged in a field or grid, on the basis of joint probability models. We will use only discrete grids, which can be an image pixel grid Λ_P or the ‘grid’ between adjacent pixels Λ_{S4}. Adjacency is here defined by 4-connectedness on the pixel grid. For correspondence estimation in Chapter 4, the correspondence field C is a field of continuous variables on Λ_P, and the discontinuity (edge-based segmentation) field is a field of discrete (binary) variables on Λ_{S4}.

In principle, the joint probability function of all stochastic variables can be used to model all possible interactions between the field variables. Since the number of variables is in the order of the number of pixels in the image (~10^6), this would lead to a tremendously complex function, both in terms of modeling and computational aspects. In MRF models all variables or grid entries interact with (depend on) each other only via their direct neighbors. Figure C.1 shows typical examples of neighborhoods on the Λ_P and Λ_{S4} lattice.

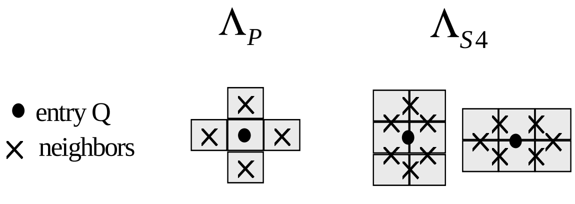


Figure C.1 Typical neighborhoods in a Markov Random Field.

In probabilistic terms, if all entries neighboring to entry Q are known, the probability distribution for the Q entry does not depend on the rest of the field:

$$p_{entry\ Q|all\ entries\ except\ Q} = p_{entry\ Q|neighbors\ of\ Q} \qquad \text{(C.1)}$$

Applying (C.1) to all entries in the field defines the joint probability for the whole field. However, a practical problem is that the joint probability is not explicitly available. This is solved by the introduction of the Gibbs Random Field (GRF), see Figure C.2. There is a one-to-one mapping between GRFs and MRFs [Gema84].

$$\begin{tikzcd}
& P_{joint} \arrow[r, leftrightarrow, "P=\frac{1}{Z}e^{-U}"] \arrow[d, leftrightarrow, "?"'] & U_{joint} \arrow[d, leftrightarrow, "\Sigma"] & \\
\text{MRF} & P_{entry|neighbors} \arrow[r, leftrightarrow, "1{:}1"'] & U_{cliques} & \text{GRF}
\end{tikzcd}$$

Figure C.2 Joint probability of MRF and GRF.

A GRF is defined in the energy domain U instead of probability p:

$$p_{MRF\,joint} = \frac{1}{Z}e^{-U_{GRF\,joint}} \tag{C.2}$$

The joint energy of the GRF is defined as a sum of clique energies:

$$U_{GRF\,joint} = \sum_{all\ cliques} U_{clique} \tag{C.3}$$

A clique is a small group of field entries whose energy is a function of the field values. The neighborhoods in MRFs are related to the cliques in GRFs. Figure C.3 shows the cliques according to the neighborhoods in Figure C.1. The neighbors of an entry Q are all entries that share a clique with Q.

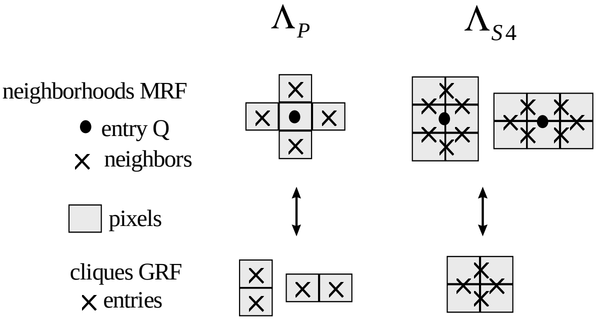


Figure C.3 Neighborhoods and cliques for Λ_P and Λ_{S4}.

The normalization constant Z in (C.2) is called the partition function and is given by (assuming a discrete valued GRF):

$$Z = \sum_{\substack{\text{all different} \\ \text{fields } f}} e^{-U_{GRF\,joint}(f)} \tag{C.4}$$

Analytical computation of Z is impossible in general, and so is numerical computation, since the space of all different fields is very high dimensional (in the order of 10^6). For the successful use of GRF models, the application should not depend on the actual value of Z.

An example of a GRF model that enforces global smoothness on a correspondence field C is:

$$U_C = \sum_{\substack{\text{all } P,Q \text{ that} \\ \text{form a clique}}} \left| C(P) - C(Q) \right|^2 \tag{C.5}$$

The cliques are chosen as depicted in Figure C.3 for the Λ_P lattice. The subtraction in (C.5) computes the first spatial derivative of the C field in x and y directions for the horizontal and vertical cliques respectively. Large variations in the C field yield a high energy U_C, which leads to a low probability for that C field, effectively smoothing the field.

Appendix D

Marker detection algorithm

Our marker detection scheme is fully automatic, extremely robust and very accurate. It measures the position of the marker centers in the images. The procedure is done separately for the multiple views of the calibration object as well as for the left and right images. It involves the following steps, illustrated by Figure D.1:

- Finding regions of interest that possibly contain a marker.
- Design of a parameterized ellipse model of a single marker inside a region.
- Estimation of the photometric parameters for each region (e.g. *SNR*).
- Detecting valid regions by a check on the photometric parameters.
- Estimation of the ellipse position within each region.
- Relating each ellipse with a marker on the plate by sorting all ellipses globally in the 8x6 grid, additionally discarding the remaining falsely detected regions.
- Estimation of the marker centers P_i by incorporating curvature effects due to lens and perspective distortion.

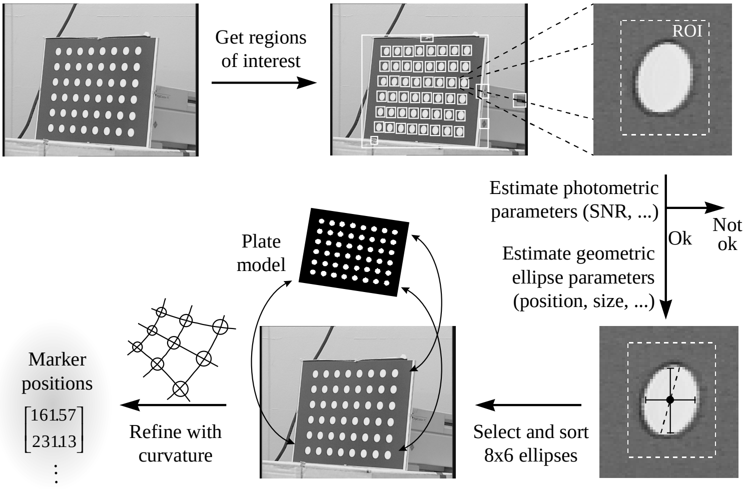

Get regions of interest
ROI
Estimate photometric parameters (SNR, ...)
Not ok
Ok
Estimate geometric ellipse parameters (position, size, ...)
Plate model
Select and sort 8x6 ellipses
Refine with curvature
Marker positions
$\begin{bmatrix} 161.57 \\ 231.13 \end{bmatrix}$
⋮

Figure D.1 The marker detection scheme.

The algorithm contains two new localization refinement methods. After the markers are roughly found by the center O_R of each region R, a well-known method for finding the marker center is the ‘center of gravity’ method, which determines the average of the image coordinates in the region weighted with luminance, here denoted by W_{lum}. Our first improvement uses the fact that the marker shape is (almost) an ellipse. We then replace the luminance weight by a refined weight $W_{ellipse}$ that is less sensitive to noise. After the markers have been sorted, we perform our second new refinement by determining the marker centers while incorporating curvature effects due to lens and perspective distortions.

In the next sections we will deal with each step one at a time.

D.1 Finding regions of interest

In this step we will search for regions that possibly contain a marker. In subsequent stages of the marker detection algorithm, we will detect if the region actually contains a marker or some other object (a false alarm). In the latter case, the region will be discarded. In this section it is thus vital that at least all markers are captured in a region, while it is less important to keep the number of false alarms low.

The idea is to use the large luminance contrast between the markers and the plate background, which produce luminance edges at the marker boundaries. When the size of the luminance gradient $|\nabla I|$ is calculated, we expect high values at the marker boundaries. When we use a simple threshold $|\nabla I| > T$, we obtain a binary image I_B which contains the marker boundaries plus, e.g., the plate border and some background objects. Then we extract 4-connected objects in I_B, of which the bounding box can serve as region, see Figure D.2. A check on the region size r_x , r_y in pixels is a simple means to discard false regions. The average diameter is defined as:

$$D = r_x + r_y \tag{D.1}$$

Then a good region is detected by:

$$D_{\min} < D < D_{\max} \tag{D.2}$$

An upper bound for $D_{\max}$ can be easily found, since eight markers plus seven inter-marker distances must be within the whole image with size N_x , N_y in pixels. This scheme is very attractive for its simplicity. However, two general features of cameras must be considered:

- Slight defocus smoothes the luminance edges. This lowers the gradient and we may miss a marker.
- Image noise is also detected by the gradient. For low noise, this introduces many small false regions, which is no problem as they are discarded later. However, when too much noise is detected, the false edges are starting to cover the image and may get 4-connected, even connecting two markers. Even if we are capable of finding one marker in such a region correctly, we will miss the other marker.

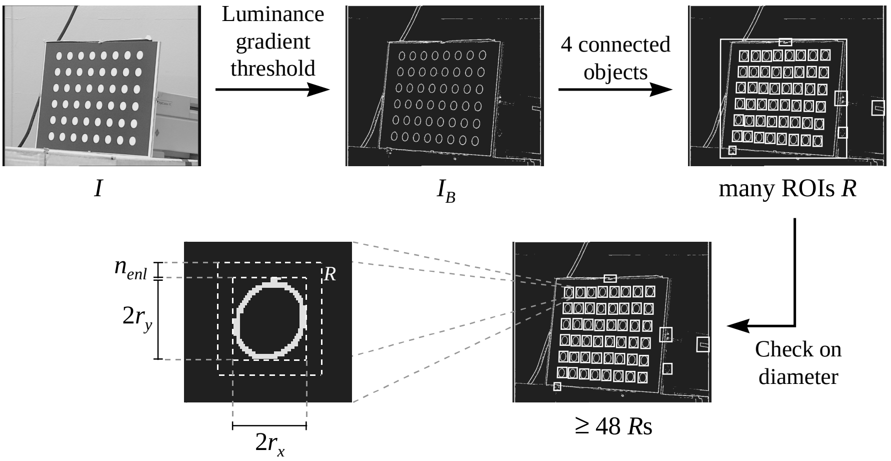


Figure D.2 Finding regions of interest by gradient thresholding and 4-connectedness.

This makes the choice of an appropriate T difficult or even impossible. Since we can easily get around these two effects, we will stick to the simple threshold scheme. The defocus effect can be circumvented by not using a [-1 1] or [-½ 0 +½] convolution filter for calculating the x gradient (and a 90° rotated version for the y gradient), but e.g. a much larger filter [-1 0 0 0 0 0 1]. As long as the filter length n_{gra} (here 7) is smaller than the ellipses themselves, but large enough to overcome the defocus smoothness, the gradient value at the edges remains approximately k. The width of the edge in the gradient image will equal n_{gra}. The image noise can be dealt with by deliberately smoothing the image. We will use a uniform filter with size n_{smo}. Whenever $n_{gra} > n_{smo}$, the gradient value at the ellipse boundary is not affected and remains approximately k.

The region of interest may be too close to the marker boundary, which may have as a result that a very small part of the marker lies outside the region. This can be counteracted by slightly enlarging the region by n_{enl} pixels. This is not necessary when a large n_{gra} is used, because then the gradient operator already yields thick boundaries.

Finally, we have to select six parameters n_{smo}, n_{gra}, T, $D_{\min}$, $D_{\max}$ and n_{enl}. Although the number of parameters seems quite high, a single set of parameters is sufficient in all practical situations (assuming an image resolution of about 500-1000, e.g. CCIR601 images):

$$\begin{bmatrix} n_{smo} \\ n_{gra} \\ T \\ D_{min} \\ D_{\max} \\ n_{enl} \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 25 \\ 5 \\ 50 \\ 1 \end{bmatrix} \tag{D.3}$$

D.2 Ellipse model of a single marker in a region

Figure D.3 shows a region R of the image I in which a single marker resides. The image luminance within the region is modeled on a pixel-by-pixel basis by:

$$I(x_R, y_R) = a + bx_R + cy_R + kW(x_R, y_R) + n(x_R, y_R) \tag{D.4}$$

For this luminance model, we introduce the region centered 2-D reference frame O_R with discrete x_R and y_R pixel coordinates. Later we will need continuous coordinates to determine the position and size of the marker with sub-pixel accuracy. Then we will use the continuous image coordinates x_I and y_I from section 2.3.6. The I and R coordinates are directly related by the position of the origin O_R of the region, which is obtained in section D.1.

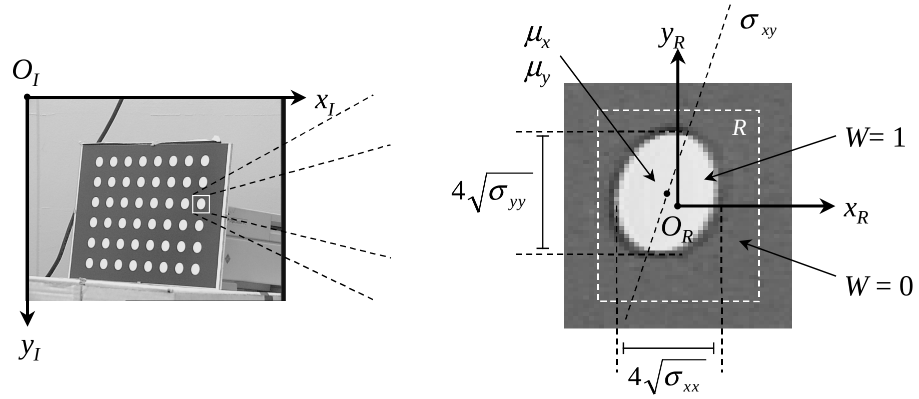


Figure D.3 The ellipse model of a single marker.

Whether a marker is present or not at a certain pixel is modeled by W. If $W = 0$, the pixel is not covered by a marker, while $W = 1$ means that the marker covers the entire pixel. Any value within [0,1] reflects the relative amount of pixel area covered. The a models the local luminance of the plate. The k is the contrast with the marker (a+k is the marker luminance). The b and c model possible non-uniform luminance in the region due to e.g. slight specular reflections of the plate and the marker. The n models image noise, which we assume to be zero mean Gaussian noise with variance σ_n, independent for each pixel in the region.

The marker presence W is modeled geometrically as follows. If circular markers are projected to images, the apparent markers will be circles when the calibration plate is parallel to the CCD (frontal view) and the lens is distortionless. If the plate is slanted with respect to the CCD, the apparent markers will be ellipses. Even under strong perspective distortion, where the closer part of the marker is projected larger than the farther part, the projection is an ellipse. If lens distortion is also included, the shape may differ from an ellipse.

We model the marker presence W by an object nearly shaped like an ellipse. With this, we allow for deformations due to lens distortion and plate manufacturing (markers are not exactly circular). A perfectly ellipse-shaped object can be described in continuous image coordinates *I* by:

$$W(x_I, y_I) = \begin{cases} 1 & r \le 1 \\ 0 & r > 1 \end{cases} \tag{D.5}$$

Here W is always 0 or 1, and r equal to:

$$r = \frac{\sigma_{yy}(x_I - \mu_x)^2 + \sigma_{xx}(y_I - \mu_y)^2 - 2\sigma_{xy}(x_I - \mu_x)(y_I - \mu_y)}{4(\sigma_{xx}\sigma_{yy} - \sigma_{xy}^2)} \tag{D.6}$$

The μ_x and μ_y are the center of the ellipse in continuous *I* coordinates. The σ_{xx} and σ_{yy} determine the size of the ellipse (see Figure D.3) and σ_{xy} its orientation. Points inside the ellipse have $r < 1$, while points outside the ellipse have $r > 1$. At the ellipse boundary we find $r = 1$. The marker projections are modeled as being almost ellipse shaped:

$$W(x_R, y_R) = \begin{cases} 1 & r \le 1 - \delta \\ 0 \ldots 1 & \text{else} \\ 0 & r \ge 1 + \delta \end{cases} \tag{D.7}$$

For each pixel x_R, y_R, we take the *I* coordinates of its center (these are integers plus one half), calculate *r* with (D.6) and then find *W* at that pixel. For *r* close to 1, we are near the ellipse boundary and *W* is fractional. For $\delta > 0$, (D.7) does not provide information about *W* in a region with width *w* around the ellipse boundary. If $w > 1$, we have spared ourselves the trouble of determining the fractional *W*s at the ellipse boundary. For larger *w*, we also incorporate the fact that the markers may not be fully ellipse shaped. Due to (D.6), the width *w* differs around the ellipse. To get an impression, for circles ($\sigma_{xx} = \sigma_{yy}$, $\sigma_{xy} = 0$), we find for small δ that $w = 2\delta\sqrt{\sigma_{xx}}$.

D.3 Estimating photometric region parameters

For each region, we estimate the model parameters *a*, *b*, *c*, *k* and σ_n in the model (D.4). We first divide the region in three areas, A_0, A_1 and $A_?$, then define a number of summations *S* on these areas, and finally estimate the parameters by combining the summations. We will now elaborate on this process.

Figure D.4 shows the areas. The A_0 area is defined as all pixels at the perimeter of the region, where we know for sure that $W = 0$. The A_1 area contains the 3x3 pixels in the center, where we know that $W = 1$. The $A_?$ is the rest (and largest part) of the region. As we are not sure about *W* in this area, we will not use it to estimate the parameters.

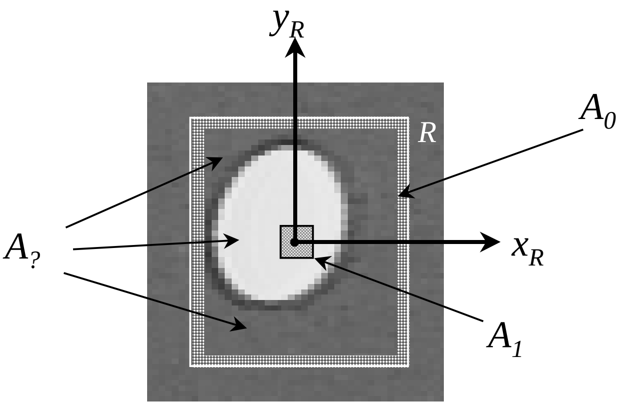


Figure D.4 Three areas in the region.

The parameters can now be estimated using specific summations over the regions. For example, we define:

$$S_{A_0} = \sum_{\substack{\text{all region}\\ \text{pixels } x_R, y_R}} A_0(x_R, y_R) \tag{D.8}$$

where A_0 is one if the pixel x_R, y_R belongs to A_0. The S_{A_0} yields the total number of pixels in that area. Similarly we define e.g. (leaving out the coordinates for simplicity):

$$S_{A_0 I x_R} = \sum_{\substack{\text{all region}\\ \text{pixels}}} A_0(x_R, y_R) I(x_R, y_R) x_R = \sum_{\substack{\text{all region}\\ \text{pixels}}} A_0 I x_R \tag{D.9}$$

This term on its own does not have much meaning, but by simple combinations we can estimate the parameters:

$$\begin{bmatrix} a \\ b \\ c \\ k+a \end{bmatrix} = \begin{bmatrix} S_{A_0 I} / S_{A_0} \\ S_{A_0 I x_R} / S_{A_0 x_R^2} \\ S_{A_0 I y_R} / S_{A_0 y_R^2} \\ S_{A_1 I} / S_{A_1} \end{bmatrix} \tag{D.10}$$

For the remaining σ_n we first construct so-called normalized luminance using (D.4):

$$I_{norm}(x_R, y_R) = I(x_R, y_R) - (a + b x_R + c y_R) \tag{D.11}$$

All variables on the right-hand side are known (image luminance I and the parameters just estimated a, b and c). In the A_0 area, the resulting I_{norm} contains only the noise. Then we find:

$$\sigma_n = \sqrt{\frac{S_{A_0 I_{norm}^2}}{S_{A_0} - 3}} \tag{D.12}$$

The minus three accounts for the fact that we have already estimated three parameters in the area A_0 (a, b and c). Then (D.12) yields an unbiased estimate of σ_n.

D.4 Detecting correct regions

We assume that specular reflection effects on the plate (non zero b and c) are reasonably small. Quantitavely we require:

$$|b| < 1 \qquad\qquad |c| < 1 \tag{D.13}$$

Further we construct the Signal-to-Noise Ratio (SNR) and require:

$$SNR = 10 \cdot {}^{10}\log \frac{\left(\frac{1}{2}k\right)^2}{\sigma_n^2} \geq \text{ 10 dB} \tag{D.14}$$

Most regions that have a perimeter with nonuniform luminance will fail to meet restrictions (D.13) and (D.14). All correct regions with markers have a perimeter that is more or less uniform and will pass the test. At this point, we check that we have ended up with at least $N_{plate\text{-}markers}$ (= 48) markers, otherwise the remaining steps are of little use.

D.5 Estimating the ellipse parameters

For the estimation of the ellipse parameters, we rely on the following which holds if the marker is perfectly ellipse-shaped given by (D.2.5) and (D.3), and if the image pixels are infinitely small (continuous image):

$$\begin{bmatrix} \mu_x \\ \mu_y \\ \sigma_{xx} + \mu_x^2 \\ \sigma_{xy} + \mu_x \mu_y \\ \sigma_{yy} + \mu_y^2 \end{bmatrix} = \begin{bmatrix} S_{Wx_I} / S_W \\ S_{Wy_I} / S_W \\ S_{Wx_I^2} / S_W \\ S_{Wx_I y_I} / S_W \\ S_{Wy_I^2} / S_W \end{bmatrix} \tag{D.15}$$

For continuous images, the summations S in (D.9) become continuous integrals. We approximate these by discrete summations like (D.9). As the coordinates used in the summation terms are still the continuous x_I , y_I coordinates, we must define which point within the pixel we use. If we use the center of the pixel, the x_I and y_I coordinates will always be an integer plus one half (see Chapter 2). In this case, the summations for the μ_x , μ_y and σ_{xy} parameters will be equivalent to continuous integrals over the entire pixel (assuming constant luminance within a pixel). The expressions for the σ_{xx} and σ_{yy} parameters will be slightly biased by 1/12, but we noticed no improvement compensating for this.

In (D.15), we need *W*, but at this moment, we only have *I*. Hence, we first construct an estimate of *W* on the basis of *I* and the estimated parameters:

$$W_{lum}(x_R, y_R) = \frac{I_{norm}(x_R, y_R)}{k} \qquad \text{(clipped to [0....1])} \tag{D.16}$$

Here we used the I_{norm} defined by (D.11), but in this case for all of the region instead of only at A_0. If luminance model (D.4) is applicable, then $W_{lum} = W$ when no noise (or luminance discretization) is present. If noise is present, it will accumulate all over the region when the summations in (D.15) are performed. If a threshold is used such as:

$$W_{thres} = \begin{cases} 0 & W_{lum} < \frac{1}{2} \\ 1 & W_{lum} \geq \frac{1}{2} \end{cases} \tag{D.17}$$

only very rare outliers due to noise will contribute, at the cost of loosing fractional *W* at the ellipse boundary. An often used compromise between (D.16) and (D.17) would be e.g. to use sigmoid thresholding, requiring the selection of the sigmoid steepness.

We will not compromise but combine the advantages of W_{lum} and W_{thres}. First we use W_{thres} to estimate preliminary ellipse shape parameters with (D.15). Then we refine the three areas A_0, A_1 and $A_?$ using the ellipse model (D.6)-(D.7), see Figure D.5. Then, we construct the final estimate of *W* that uses the ellipse shape:

$$W_{ellipse} = A_1 + A_? W_{lum} \tag{D.18}$$

In this way, we used W_{thres}, which is insensitive to noise, in most of the region (due to its effect on the area refinement), and W_{lum}, which contains the fractional marker presence, only at the ellipse boundary. With (D.15) working on $W_{ellipse}$ instead of *W* we find the final ellipse parameters.

The δ in (D.7) regulates *w*, which is the size of $A_?$, in which W_{lum} is active via (D.18). It can be seen as some sort of geometrical equivalent of the photometric sigmoid steepness parameter discussed above. We found that when δ are used that produce $A_?$ with a thickness of about $w \approx 4$ pixels, the results are best. For circular markers, this can be accomplished by:

$$\delta = \frac{2}{\sqrt{\sigma_{xx}}} \tag{D.19}$$

For ellipse shapes, the results are similar. Figure D.5 shows the complete procedure for the ellipse parameter estimation.

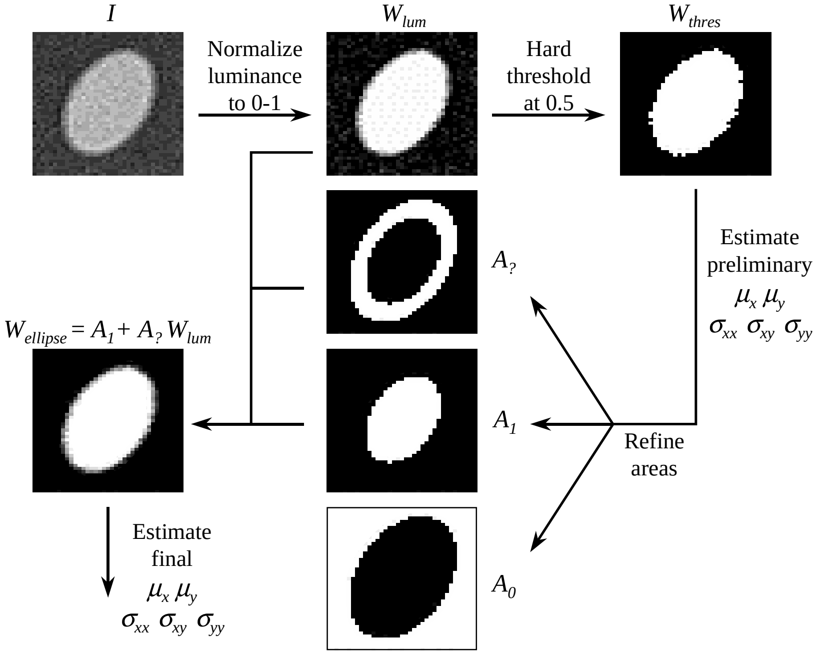

I
W_{lum}
W_{thres}
Normalize luminance to 0-1
Hard threshold at 0.5
Estimate preliminary
$\mu_x\ \mu_y$
$\sigma_{xx}\ \sigma_{xy}\ \sigma_{yy}$
$A_?$
$W_{ellipse} = A_1 + A_? W_{lum}$
A_1
Refine areas
A_0
Estimate final
$\mu_x\ \mu_y$
$\sigma_{xx}\ \sigma_{xy}\ \sigma_{yy}$

Figure D.5 The ellipse parameter estimation procedure.

At this point, we have determined the position of at least 48 objects, which we will call the ellipses Q_j. If everything went well, 48 of those ellipses correspond to markers. For those ellipses, the found position parameters localize a marker P_i already quite accurate.

D.6 Linking ellipses with markers in the grid

In this step, we find the 48 markers among all ellipses Q_j, and arrange them in an 8x6 grid to link them with the marker centers P_i of the calibration plate. The method is the following. We construct all possible rows of eight ellipses and select the best six according to some criterion. Within each row, we sort on x_I position to get the column position c in [0,7]. Then we sort the rows by taking the y_I position of the first markers of each row. This yields the row number r. Now all ellipses Q_j have a row number r_j and a column number c_j. The correspondence is then made by:

$$Q_j \leftrightarrow P_{c_j+8r_j} \qquad \text{(D.20)}$$

We select the six best rows of eight ellipses from all found ellipses as follows. First we construct all possible rows L of eight markers, and choose the best with:

$$L_{best} = \arg \min_{\substack{\text{all possible rows} \\ L \text{ of 8 markers}}} U_{row}(L) \tag{D.21}$$

Although the number of possible rows seems extremely large, this minimization can be implemented very efficiently by means of a backtracking algorithm. The U_{row} function is defined as:

$$U_{row}(L) = \sum_{\substack{\text{all six triples} \\ \text{of consecutive} \\ \text{ellipses } Q_i, Q_j, Q_k \\ \text{in row } L}} U_{triple}\left(Q_i, Q_j, Q_k\right) \tag{D.22}$$

The U_{triple} function measures the second derivative of both photometric and geometric ellipse parameters:

$$\begin{aligned} U_{triple}\left(Q_i, Q_j, Q_k\right) = {} & \left(a_i - 2a_j + a_k\right)^2 + \left(k_i - 2k_j + k_k\right)^2 + \\ & \left(\mu_{x;i} - 2\mu_{x;j} + \mu_{x;k}\right)^2 + \left(\mu_{y;i} - 2\mu_{y;j} + \mu_{y;k}\right)^2 + \\ & \left|\sigma_{xx;i} - 2\sigma_{xx;j} + \sigma_{xx;k}\right| + \left|\sigma_{xy;i} - 2\sigma_{xy;j} + \sigma_{xy;k}\right| + \left|\sigma_{yy;i} - 2\sigma_{yy;j} + \sigma_{yy;k}\right| \end{aligned} \tag{D.23}$$

This function yields minimal U_{row} for rows with neighboring ellipses in straight lines, with similar interdistances, sizes, shapes and luminances. The σ parameters are not squared, since their definition already includes a square.

The method only works when the number of markers per column (= 8) is larger than that per row (= 6). If we use the columns in (D.21) instead of rows, columns may be found that are actually part of a row or a diagonal. The dark background of the plate is large enough so that no background object may form a competing row with seven markers from a real row.

Whenever a row is found, the eight ellipses are put aside and the algorithm is repeated with the remaining ellipses. This continues until six rows are found. Since the ellipse centers Q are now in correspondence with the markers, we will use the same index i for both:

$$Q_i \leftrightarrow P_i \tag{D.24}$$

D.7 Perspective and lens distortion

The 2-D image point that we are trying to locate is the projection of the point P_i, which is defined as the center of the marker in 3-D or *CF* coordinates. This may not coincide with the ellipse center Q_i due to perspective and lens distortion [Heik97], see Figure D.6. We will model this effect by taking into account curvature between the *CF* (calibration plate) and *I* (image) coordinates. Note that unlike lens distortion, perspective distortion does not yield curved axes. It does change the length scale between *CF* and *I* coordinates with image position, which is also called curvature mathematically.

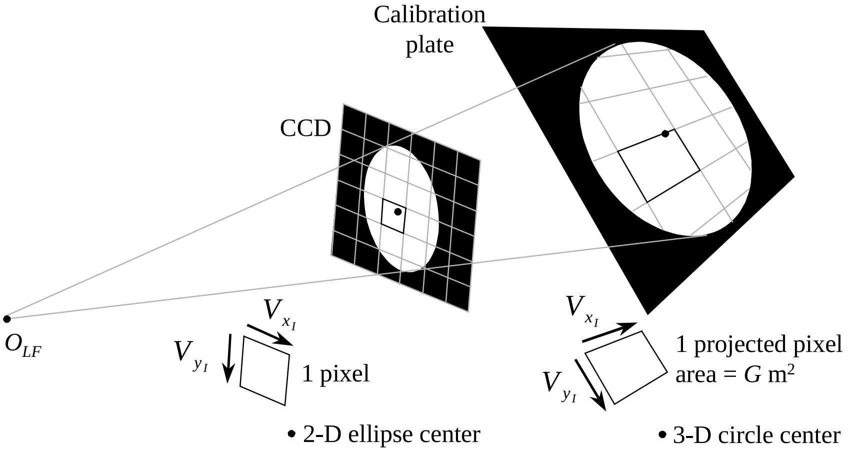


Figure D.6 Curvature effects due to lens and perspective distortion make each CCD pixel project to a different area on the calibration plate. This makes the 2-D center of the ellipse (projected marker) differ from the projection of the 3-D circle (real marker) center.

The idea behind our approach is to determine how much 2-D marker area *G* measured in *CF* calibration plate coordinates (meters) corresponds to each single pixel in the ellipse. We can then recalculate (D.15) with (D.18) and *G*. At this point, we are only interested in refining the position:

$$\begin{bmatrix}\mu_x \\ \mu_y\end{bmatrix} = \begin{bmatrix} S_{GW_{ellipse}x_I} / S_{GW_{ellipse}} \\ S_{GW_{ellipse}y_I} / S_{GW_{ellipse}} \end{bmatrix} \tag{D.25}$$

If the size and shape parameters were also calculated, they would have meaning for the actual markers on the plate. We would thus find that $\sigma_{xy} \approx 0$ and $\sigma_{xx} \approx \sigma_{yy}$, which hold for circles. Via (D.25) we find an unbiased estimate of the projected 3-D center of the marker. For the marker area *G* we find the following using Appendix A:

$$G = V_{x_I} \times V_{y_I} \ \text{(in 2-D frame } CF) = \varepsilon_{\sigma_{CF},\tau_{CF}} V_{x_I}^{\sigma_{CF}} V_{y_I}^{\tau_{CF}} = V_{x_I}^{x_{CF}} V_{y_I}^{y_{CF}} - V_{x_I}^{y_{CF}} V_{y_I}^{x_{CF}} \tag{D.26}$$

At the right-hand side, we find the base matrix $V_{\sigma_I}^{\sigma_{CF}}$. In this case only the *x* and *y* coordinates are used in both the *CF* and *I* frames. The 2-D spaces of interest are the calibration plate surface $z_{CF} = 0$ and the CCD plane $z_I = 0$. If the base matrix is a constant with respect to image position, there is no curvature. In that case, *G* is a constant and the result from (D.25) is the same as the earlier result for the ellipse center. This happens when the plate and CCD are parallel, and there is no lens distortion.

Assuming that the base matrix changes only slowly with image position, we can denote its dependency on image position in a small region around each marker position P_i by a linear equation:

$$V_{\sigma_I}^{\sigma_{CF}}\left(P_i+\Delta P\right)=V_{\sigma_I}^{\sigma_{CF}}\left(P_i\right)+\Delta P^{\tau_I}V_{\sigma_I,\tau_I}^{\sigma_{CF}}\left(P_i\right) \tag{D.27}$$

The ΔPs refer to all points in the region around marker point P_i. At the right-hand side of (D.27), the curvature tensor $V_{\sigma_I,\tau_I}^{\sigma_{CF}}$ appears as defined in Appendix A. To use (D.27) in (D.26), we must estimate the both base matrix and the curvature tensor at each P_i. For this, we first integrate (D.27):

$$\left(P_i+\Delta P\right)^{\sigma_{CF}}=P_i^{\sigma_{CF}}+\Delta P^{\sigma_I}V_{\sigma_I}^{\sigma_{CF}}\left(P_i\right)+\tfrac{1}{2}\Delta P^{\sigma_I}\Delta P^{\tau_I}V_{\sigma_I,\tau_I}^{\sigma_{CF}}\left(P_i\right) \tag{D.28}$$

and then use neighboring marker points P_j around P_i as ΔP:

$$P_j^{\sigma_{CF}}=P_i^{\sigma_{CF}}+\left(P_j^{\sigma_I}-P_i^{\sigma_I}\right)V_{\sigma_I}^{\sigma_{CF}}\left(P_i\right)+\tfrac{1}{2}\left(P_j^{\sigma_I}-P_i^{\sigma_I}\right)\left(P_j^{\tau_I}-P_i^{\tau_I}\right)V_{\sigma_I,\tau_I}^{\sigma_{CF}}\left(P_i\right) \tag{D.29}$$

For each P_i, we will use a 3x3 neighborhood of markers P_j closest to P_i. Except for the markers at the border of the plate, the marker of interest lies in the center of the 9 markers. Here, we will consider the latter situation, shown in Figure D.7.

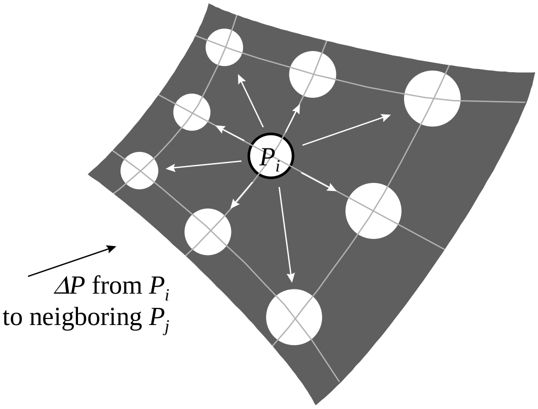


Figure D.7 For each marker, a 3x3 neighborhood is constructed to estimate the local curvature around the marker.

Since we do not know the I coordinates of the markers P_i, we must resort to taking the ellipse centers Q_i as approximation to the marker centers:

$$\left(P_j^{\sigma_{CF}}-P_i^{\sigma_{CF}}\right)\approx\left(Q_j^{\sigma_I}-Q_i^{\sigma_I}\right)V_{\sigma_I}^{\sigma_{CF}}\left(Q_i\right)+\tfrac{1}{2}\left(Q_j^{\sigma_I}-Q_i^{\sigma_I}\right)\left(Q_j^{\tau_I}-Q_i^{\tau_I}\right)V_{\sigma_I,\tau_I}^{\sigma_{CF}}\left(Q_i\right) \tag{D.30}$$

At this point, the only unknowns in (D.30) are the base matrix and the curvature tensor. The former contains 4 parameters and the latter 6 (see Appendix A). For $j = i$, the equation is trivial, but for each of the other eight neighbors we have two equations (one for the x_{CF} component and one for y_{CF}). Since (D.30) is linear in the base matrix and curvature tensor, we can find them easily with a least-squares estimation technique using the 16 equations.

In the left-hand side of (D.30) the *CF* coordinates of the markers are present. Thus it seems that this part of the algorithm needs the full geometric calibration plate model. However, it does not require e.g. the 10 μm precision that our professional A1 sized plate has (see Table 3.1). We can approximate the plate model by a perfectly rectangular grid. The relative errors are then e.g. in the order of 10^{-3} (the markers on the A1 plate are in an exact grid up to about 1 mm), so they are completely negligible with respect to the off-grid effects due to curvature as shown in Figure D.7. Also, since (D.25) is invariant with respect to scale changes in *G*, the absolute scale of the grid need not be known. If we take the row and column numbers as length units, we can use the following model for any plate similar to the A1 and A4 plates from Table 3.1:

$$\left(P_j^{\sigma_{CF}} - P_i^{\sigma_{CF}}\right) \approx \begin{bmatrix} c_j - c_i \\ r_j - r_i \end{bmatrix} \tag{D.31}$$

In order to compensate for the approximation from (D.29) to (D.30) we perform the curvature refinement algorithm two times: first with the ellipse centers, and then with the marker positions obtained in the first run.

Bibliography

[Acca95] M. Accame, F.G.B. De Natale and D.D. Giusto, “Hierarchical block matching for disparity estimation in stereo sequences”, *Proceedings of ICIP95*, pp. 374-377, 1995

[Ariy98] A.M. Ariyaeeinia, “Analysis and design of stereoscopic television systems”, in *Signal Processing: Image Communication*, Vol. 13, pp. 201-208, 1998

[Arms96] M.N. Armstrong, “Self-calibration from image sequences”, *Ph.D. Thesis*, Department of Engineering Science, University of Oxford, 1996

[Azar95] A. Azarbayejani and A.P. Pentland, “Recursive estimation of motion, structure, and focal length”, *IEEE Transactions on PAMI*, Vol. 17, No. 6, pp. 562-575, 1995

[Basu95] A. Basu, “Active calibration of cameras: theory and implementation”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 2, pp. 256-265, 1995

[Barn80] S.T. Barnard and W.B. Thompson, “Disparity analysis of images”, *IEEE Transactions on PAMI*, Vol. 2, No. 4, pp. 333-340, 1980

[Bert88] M. Bertero, T. Poggio and V. Torre, “Ill-posed problems in early vision”, *Proceedings of the IEEE*, Vol. 76, No. 8, pp. 869-889, 1988

[Biem87] J. Biemond, L. Looijenga, D.E. Boekee and R.H.J.M. Plompen, “A pel-recursive Wiener-based displacement estimation algorithm”, *Signal Processing*, Vol. 13, No. 4, pp. 399-412, 1987

[Börö91] L. Böröczky. “Pel-recursive motion estimation”, *Ph.D. thesis*, Delft University of Technolgy, 1991

[Boug98] S. Bougnoux, “From projective to Euclidean space under any practical situation, a criticism of self-calibration”, *Proc. of ICCV*, pp. 790-796, 1998

[Boye94] K.L. Boyer and S. Sarkar, “On the localization performance measure and optimal edge detection”, *IEEE Transactions on PAMI*, Vol. 16, No. 1, pp. 106-110, 1994

[Brai95a] J.C. Brailean and A.K. Katsaggelos, “A recursive nonstationary MAP displacement vector field estimation algorithm”, *IEEE Transactions on Image Processing*, Vol. 4, No. 4, pp. 416-429, 1995

[Brai95b] J.C. Brailean and A.K. Katsaggelos, “Simultaneous recursive displacement estimation and restoration of noisy-blurred image sequences”, *IEEE Transactions on Image Processing*, Vol. 4, No. 9, pp. 1236-1251, 1995

[Brow71] D.C. Brown, “Close-range camera calibration”, *Photogrammetric Engineering*, Vol. 37, pp. 855-866, 1971

[Cann86] J. Canny, “A computational approach to edge detection”, *IEEE Transactions on PAMI*, Vol. 8, No. 6, pp. 679-698, 1986

[Chan94] M.M. Chang, M.I. Sezan and A.M. Tekalp, “An algorithm for simultaneous motion estimation and scene segmentation”, *in Proceedings of ICASSP94*, No. 5, pp. 221-224, 1994

[Chup94] B. Chupeau and P. Salmon, “Synthesis of intermediate pictures for autostereoscopic multiview displays”, in *Proceedings of the workshop on HDTV, Turin, Italy*, 1994

[Ceo] Compton’s Encyclopedia Online, http://www.comptons.com

[Cox95] I.J. Cox, S. Roy and S. Hingorani, “Dynamic histogram warping of image pairs for constant image brightness”, *Proceedings of ICIP95*, pp. 366-369, 1995

[Cox96] I.J. Cox, S.L. Hingorani and S.B. Rao, “A maximum likelihood stereo algorithm”, *Computer Vision and Image Understanding*, Vol. 63, No. 3, pp. 542-567, 1996

[Cruz93] C. Cruz-Neira, D.J. Sandlin and T.A. DeFanti, “Surround-screen projection-based virtual reality: The design and implementation of the CAVE”, *SIGGRAPH* 1993

[Csur97] G. Csurka, C. Zeller, Z. Zhang and O.D. Faugeras, “Characterizing the uncertainty of the fundamental matrix”, *Computer Vision and Image Understanding*, Vol. 68, No. 1, pp. 18-36, 1997

[Davi92] J. Davidse, “Televisietechniek en beeldversterking”, Delftse Uitgevers Maatschappij, 1992

[Deve96] F. Devernay and O. Faugeras, “From projective to Euclidean reconstruction”, *Proc. Conf. On Computer Vision and Pattern Recognition (CVPR)*, 1996

[Dist95] DISTIMA, European RACE 2045 project, http://www.tnt.uni-hannover.de /project/eu/distima, 1992-1995

[Drie92] J.N. Driessen, “Motion estimation for digital video”, *Ph.D. thesis*, Delft University of Technology, 1992

[Ebo] Encyclopedia Britannica Online, http://www.eb.com

[Eela99a] I. van den Eelaart and E.A. Hendriks, "A flexible camera calibration system that uses straight lines in a 3D scene to calculate the lens distortion", *Proc. of the 5th annual conference of the Advanced School for Computing and Imaging (ASCI'99)*, pp. 443-448, 1999.

[Eela99b] I. van den Eelaart, "The design of a camera calibration system", *Research report*, ICT Group, Delft University of Technology, 1999

[Faug92] O.D. Faugeras, Q.T. Luong and S.J. Maybank, "Camera self-calibration: theory and experiments", *Proc. of 2nd European Conference on Computer Vision*, pp. 321-334, 1992

[Faug93] O. Faugeras, "Three-dimensional computer vision, a geometric viewpoint", MIT Press, 1993

[Faug96] O. Faugeras, L. Robert, "What can two images tell us about a third one?", in *International Journal of Computer Vision*, Vol. 18, No. 1, pp. 5-20, 1996

[Faug97] O. Faugeras and T. Papadopoulo, "A nonlinear method for estimating the projective geometry of three views", *INRIA research report*, No. 3221, 1997

[Fitz98] A. Fitzgibbon and A. Zisserman, "Automatic 3D model acquisition and generation of new images from video sequences", in *Proc. EUSIPCO*, pp. 1261-1269, 1998

[Fran96] R.E.H. Franich, "Disparity estimation in stereoscopic digital images", *Ph.D. thesis*, Delft University of Technology, 1996

[Fran95] R.E.H. Franich, R.L. Lagendijk and J. Biemond, "A disparity space image path", *Proceedings of the International Workshop on Three Dimensional Imaging (IWS3DI'95)*, pp. 122-127, 1995

[Fuji96] T. Fujii, T. Kimoto and M. Tanimoto, "Ray space coding for 3D visual communication", in *Proc. Picture Coding Symposium*, Vol. 2, pp. 447-451, 1996

[Gema84] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", *IEEE Transactions on PAMI*, Vol. 6, No. 6, pp. 721-741, 1984

[Gisc93] V.A. Giscard d'Estaing and M. Young, "Inventions and discoveries 1993", Facts on File Inc, New York, 1993

[Gram98] N. Grammalidis and M.G. Strintzis, "Disparity and occlusion estimation in multiocular systems and their coding for the communication of multiview image sequences", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, No. 3, pp. 328-344, 1998

[Grin94] V.S. Grinberg, G. Podnar and M.W. Siegel, "Geometry of binocular imaging", in *Proc. SPIE Conference on Stereoscopic Displays and Applications V*, 1994

[Haan92] G. de Haan, "Motion estimation and compensation", *Ph.D. thesis*, Delft University of Technology, 1992

[Heik97] J. Heikkilä and O. Silvén, “A four-step camera calibration procedure with implicit image correction”, *Proc. Conf. On Computer Vision and Pattern Recognition (CVPR),* pp. 1106-1112, 1997

[Hein] Heinrich-Hertz-Institut, human factors department, “3D-displays”, http://atwww.hhi.de/3D_displays

[Heit93] F. Heitz and P. Bouthemy, “Multimodal estimation of discontinuous optical flow using Markov random fields”, *IEEE Transactions on PAMI*, Vol. 15, No. 2, pp. 1217-1232, 1993

[Hend96] E.A. Hendriks and Gy. Marosi, “Recursive disparity estimation algorithm for real time stereoscopic video applications”, *Proceedings of ICIP96*, pp. 891-894, 1996

[Herm71] S. Herman, “Principles of binocular 3D displays with applications to television”, in *Journal of the SMPTE*, Vol. 80, pp. 539-544, 1971

[Hmd] Overview of Head Mounted Displays, http://www.stereo3d.com/hmd.htm

[Horn86] B.K.P. Horn, “Robot vision”, MIT Press, McGraw-Hill Book Company, 1986

[Inti94] S.S. Intille and A.F. Bobick, “Disparity-space images and large occlusion stereo*”, MIT Media Lab Perceptual Coding Group*, Technical Report No. 220, 1994

[Jeba99] T. Jebara, A. Azarbayejani and A. Pentland, “3D structure from 2D motion”, in *IEEE Signal Processing Magazine, special issue on 3D and stereoscopic visual communication*, Vol. 16, No. 3, pp. 66-84, 1999

[Jong99] C. Jongeneel, “Televisie van meerdere kanten bekeken”, *Delft Integraal*, No. 6, pp. 12-15, 1999

[Kaji97] Y. Kajiki, H. Yoshikawa and T. Honda, “Hologram-like video images by 45-view stereoscopic display”, in *Proc. of Conf. on Stereoscopic Displays and Virtual Reality Systems IV*, SPIE Vol. 3012, pp. 154-166, 1997

[Kana94] T. Kanade and M. Okutomi, “A stereo matching algorithm with an adaptive window: theory and experiment”, *IEEE Transactions on PAMI*, Vol. 16, No. 9, pp. 1207-1212, 1994

[Kang99] S.B. Kang, “A survey of image-based rendering techniques”, in *SPIE Proceedings*, Vol. 3641, pp. 2-16, 1999

[Konr92] J. Konrad and E. Dubois, “Bayesian estimation of motion vector fields*”, IEEE Transactions on PAMI*, Vol. 14, No. 9, pp. 910-927, 1992

[Konr99] J. Konrad, “Enhancement of viewer comfort in stereoscopic viewing: parallax adjustment”, in *SPIE proceedings*, Vol. 3639, pp. 179-190, 1999

[Kutk94] R. Kutka, “Reconstruction of correct 3-D perception on screens viewed at different distances”, in *IEEE Transactions on Communications*, Vol. 42, No. 1, pp. 29-33, 1994

[Levo96] M. Levoy and P. Hanrahan, “Light field rendering”, *ACM SIGGRAPH*, pp. 31-42, 1996

[Li92] S.Z. Li, “On discontinuity-adaptive smoothness priors in computer vision”, *IEEE Transactions on PAMI*, Vol. 17, No. 6, pp. 576-586, 1995

[Liu93] J. Liu and R. Skerjanc, “Stereo and motion correspondence in a sequence of stereo images”, *Signal Processing: Image Communication*, Vol. 5, pp. 305-318, 1993

[Liu95] J. Liu, I.P. Beldie and M. Wöpking, “A computational approach to establish eye-contact in videocommunication”, in *Proc. of the International Workshop on Three Dimensional Imaging (IWS3DI95)*, pp. 229-234, 1995

[Long81] H.C. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections”, *Nature*, Vol. 293, No. 10, pp. 133-135, 1981

[Luce97] M. Lucente, “Interactive three-dimensional holographic displays: seeing the future in depth”, in *SIGGRAPH Computer Graphics, special issue on Current, New and Emerging Display Systems*, May 1997

[Luet93] M.R. Luettgen, W.C. Karl, A.S. Willsky and R.R. Tenney, “Multiscale representations of Markov Random Fields”, *IEEE Transactions on PAMI*, Vol. 41, No. 12, pp. 3377-3396, 1993

[Lund98] A. Lundmark, N. Wadström and H. Li, “Hierarchical subsampling giving fractal regions”, Technical Report ISSN 1400-3902, LiTH-ISY-R-2035, Linköping University, Sweden, 1998

[Luon93] Q.T. Luong and O. Faugeras, “Self-calibration of a stereo rig from unknown camera motions and point correspondences”, *INRIA research report*, No. 2014, 1993

[Mara89] P. Maragos, “Morphological correlation and mean absolute error criteria”, *Proceedings of ICASSP’89*, pp. 1568-1571, 1989

[Matt89] L. Matthies, T. Kanade and R. Szeliski, “Kalman filter-based algorithms for estimating depth from image sequences”, *International Journal of Computer Vision*, Vol. 3, pp. 209-238, 1989

[Nage86] H.H. Nagel and W. Enkelmann, “An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences”, *IEEE Transactions on PAMI*, Vol. 8, No. 5, pp. 565-593, 1986

[Ohm97] J.R. Ohm and E. Izquierdo, “An object-based system for stereoscopic viewpoint synthesis”, in *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 5, pp. 801-811, 1997

[Ohm98] J.R. Ohm, K. Grueneberg, E. Hendriks, M.E. Izquierdo, D. Kalivas, M. Karl, D. Papadimatos and A. Redert, “A realtime hardware system for stereoscopic videoconferencing with viewpoint adaptation”, in *Signal Processing: Image Communication*, Vol. 14, pp. 147-171, 1998

[Ohta85] Y. Ohta and T. Kanade, “Stereo by intra- and inter-scanline search using dynamic programming”, *IEEE Transactions on PAMI*, Vol. 7, No. 2, pp. 139-154, 1985

[Orig] Origin Systems Corporation, http://www.orin.com

[Pano98a] PANORAMA, European ACTS AC092 project, http://www.tnt.uni-hannover.de /project/eu/panorama/, 1995-1998

[Pano98b] PANORAMA Final Report, internal report to EU commission, Brussels, 1998

[Pano98c] PANORAMA deliverable D28, internal report to EU commission, Brussels, 1998

[Papa95] D.V. Papadimitriou and T.J. Dennis, "Epipolar line estimation and rectification for stereo image pairs*"*, *Proceedings of the International Workshop on Three Dimensional Imaging (IWS3DI'95)*, pp. 128-133, 1995

[Pasm97] W. Pasman, "Enhancing x-ray baggage inspection by interactive viewpoint selection" , *Ph.D. thesis*, Delft University of Technology, 1997

[Pasm99] W. Pasman, A. van der Schaaf, R.L. Lagendijk and F.W. Jansen, "Low latency rendering for mobile augmented reality", *Proc. of the 5th annual conference of the Advanced School for Computing and Imaging (ASCI'99)*, pp. 372-376, 1999

[Past91] S. Pastoor, "3D-television: A survey of recent research results on subjective requirements", in *Signal Processing: Image Communication*, Vol. 4, pp. 21-32, 1991

[Patr97] I. Patras, E.A. Hendriks and G. Tziritas, "Construction of multiple views using jointly estimated motion and disparity fields", *Proceedings of VCIP97*, SPIE Vol. 3024, pp. 380-387, 1997

[Pede95] F. Pedersini, A. Sarti and S. Tubaro, "Synthesis of virtual views using non-Lambertian reflectivity models and stereo matching", In Proceedings of *ICIP95*, Vol. 2, pp. 358-361, 1995

[Pede97a] F. Pedersini, D. Pele, A. Sarti and S. Tubaro, "Calibration and self-calibration of multi-ocular camera systems", in proceedings of the *International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging* (IWSNHC3DI'97), Rhodos, Greece, pp. 81-84, 1997

[Pede97b] F. Pedersini, A. Sarti and S. Tubaro, "3D surface reconstruction from horizons", *Proceedings of the International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging, (IWSNHC3DI'97)*, pp. 85-88, 1997

[Pede99] F. Pedersini, A. Sarti and S. Tubaro, "Multi-camera systems: calibration and applications", in *IEEE Signal Processing Magazine, special issue on 3D and stereoscopic visual communication*, Vol. 16, No. 3, pp. 55-65, 1999

[Phil] Philips lenticular screens, http://www.research.philips.com/generalinfo /special/3dlcd

[Poll98] M. Pollefeys, R. Koch, M. Vergauwen and L. Van Gool, "Flexible acquisition of 3D structure from motion", in *Proceedings of the IEEE Image and Multidimensional Digital Signal Processing (IMDSP) Workshop '98*, pp. 195-198, 1998

[Pres92] W.H. Press, S. A. Teukolsky, W.T. Vetterling and B.P. Flannery, “Numerical recipes in C, the art of scientific computing”, Cambridge University Press, 1992

[Rede97a] P.A. Redert and E.A. Hendriks, “Disparity map coding for 3d teleconferencing applications”, *Proc. Conference on Visual Communications and Image Processing (VCIP)*, SPIE Vol. 3024, pp. 369-379, 1997

[Rede97b] P.A. Redert, E.A. Hendriks and J. Biemond, “Synthesis of multi viewpoint images at non intermediate positions”, In *Proceedings of ICASSP97*, pp. 2749-2752, 1997

[Rede97c] P.A. Redert and E.A. Hendriks, “A new disparity map format for 3d teleconferencing applications”, *Proc. of the 3rd annual conference of the Advanced School for Computing and Imaging (ASCI'97)*, pp. 266-271, 1997

[Rede97d] P.A. Redert and E.A. Hendriks, “An efficient disparity map format for real time interpolation in multi viewpoint stereoscopic video systems”, *Proc. of the International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging, (IWSNHC3DI'97)*, pp. 155-158, 1997

[Rede97e] P.A. Redert, J.J. van Klaveren and E.A. Hendriks, “Accurate 3D eye tracking for multi viewpoint systems”, *Proc. of the International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging (IWSNHC3DI)*, pp. 224-227, 1997

[Rede97f] P.A. Redert, “Acquisition and presentation of 3D video in the PANORAMA hardware chain”, PANORAMA report to EU commission, 1997

[Rede98a] P.A. Redert, C.J. Tsai, E.A. Hendriks, and A.K. Katsaggelos, “Disparity estimation with modeling of occlusion and object orientation”, *Proc. of VCIP98*, SPIE volume 3309, pp. 798-808, 1998

[Rede98b] P.A. Redert and E.A. Hendriks “ Disparity estimation by searching parallel and orthogonal to the epipolar lines”, *Proc. of the 4th annual conference of the Advanced School for Computing and Imaging (ASCI'98*), pp. 184-188, 1998

[Rede98d] P.A. Redert and E.A. Hendriks, “Self calibration of stereo cameras with lens distortion”, *Proc. of the IEEE Image and Multidimensional Digital Signal Processing (IMDSP) Workshop '98,* pp. 163-166, 1998

[Rede99a] P.A. Redert, E.A. Hendriks and J.Biemond, “Correspondence estimation in image pairs”, *IEEE Signal Processing Magazine, special issue on 3D and stereoscopic visual communication*, Vol. 16, No. 3, pp. 29-46, 1999

[Rede99b] P.A. Redert and E.A. Hendriks, “Self-calibration of stereo cameras with a probabilistic camera model including lens distortion”, *Proc. of the 5th annual conference of the Advanced School for Computing and Imaging (ASCI'99)*, pp. 144-149, 1999

[Rede99c] P.A. Redert, B. Kaptein, M.J.T. Reinders, I. Van den Eelaart and E.A. Hendriks, “Extraction of semantic 3D models of human faces from stereoscopic image sequences”, *Acta Stereologica*, Vol. 18, No. 2, pp. 255-264, 1999

[Rede99d] P.A. Redert and E.A. Hendriks, "Acquisition of 3-D scenes with a single hand held camera", accepted for publication in *Proc. Int. Workshop on Synthetic - Natural Hybrid Coding and Three Dimensional Imaging (IWSNHC3DI)*, 1999.

[Rede00] P.A. Redert, E.A. Hendriks and J. Biemond, "3-D Scene reconstruction with viewpoint adaptation on stereo displays", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 4, pp. 550-562, 2000

[Reyn96] D.P. McReynolds and D.G. Lowe, "Rigidity checking of 3D point correspondences under perspective projection", *IEEE Transactions on PAMI*, Vol. 18, No. 12, pp. 1174-1185, 1996

[Roth97] C. Rothwell, O. Faugeras and G. Csurka, "A comparison of projective reconstruction methods for pairs of views", *Computer Vision and Image Understanding*, Vol. 68, No. 1, pp. 37-58, 1997

[Roy97] S. Roy, J. Meunier and I.J. Cox, "Cylindrical rectification to minimize epipolar distortion", *Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 393-399, 1997

[Schu85] B.F. Schutz, "A first course in general relativity", Cambridge University Press, 1985

[Seit95] S.M. Seitz and C.R. Dyer, "Physically-valid view synthesis by image interpolation", *Proc. IEEE Workshop on representation of visual scenes*, 1995

[Sext99] I. Sexton and P. Surman, "Stereoscopic and autostereoscopic display systems", in *IEEE Signal Processing Magazine, special issue on 3D and stereoscopic visual communication*, Vol. 16, No. 3, pp. 85-99, 1999

[Slam80] C.C. Slama (ed.), "Manual of photogrammetry", 4th edition, *American Society of Photogrammetry*, 1980

[Snyd91] M.A. Snyder, "On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction", *IEEE Transactions on PAMI*, Vol. 13, No. 11, pp. 1105-1114, 1991

[Stei97a] G.P. Stein, "Lens distortion calibration using point correspondences", in *IEEE Conference on CVPR*, pp. 602-609, 1997

[Stei97b] E. Steinbach, S. Chaudhuri and B. Girod, "Robust estimation of multi-component motion in image sequences using the epipolar constraint", In *Proceedings of ICASSP97*, pp. 2689-2692, 1997

[Ster] Stereographics products, http://www.stereographics.com

[Stil97] C. Stiller, "Object-based estimation of dense motion fields", *IEEE Transactions on Image Processing*, Vol. 6, No. 2, pp. 234-250, 1997

[Stri99] M.G. Strintzis and S. Malassiotis, "Object-based coding of stereoscopic and 3D image sequences", *IEEE Signal Processing Magazine, special issue on 3D and stereoscopic visual communication*, Vol. 16, No. 3, pp. 14-28, 1999

[Teka95a] A.M. Tekalp, "Digital Video Processing", Prentice Hall, 1995

[Teka95b] Section 8.3 of [Teka95a]

[Ther92] C.W. Therrien, “Discrete random signal signals and statistical signal processing”, Prentice Hall, 1992

[Truc98] E. Trucco and A. Verri, “Introductory techniques for 3-D computer vision”, Prentice Hall, 1998

[Tsai87] R.Y. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses”, *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, 1987

[Tsai97] C.J. Tsai and A.K. Katsaggelos, “Optical flow estimation for multi-channel video sequences”, *Proceedings of VCIP97*, SPIE Vol. 3024, pp. 360-368 , 1997

[Tsen95] B.L. Tseng and D. Anastassiou, “A theoretical study on an accurate reconstruction of multiview images based on the Viterbi algorithm”, *Proceedings of ICIP95*, pp. 378-381, 1995

[Turi] The Turing Institute, The history of stereo photography, http://www.turing.gla.ac.uk/funsite/history.htm

[Tzov96] D. Tzovaras, N. Grammalidis and M.G. Strintzis, "Disparity field and depth map coding for multiview image sequence compression”, *Proc. of the International Conference on Image Processing (ICIP)*, pp. 887-890, 1996

[Vais99] L. Vaissie, J.P. Rolland and G.M. Bochenek, “Analysis of eyepoint locations and accuracy of rendered depth in binocular head-mounted displays”, in *Proc. Conference on Stereoscopic Displays and Applications X*, SPIE Vol. 3639, pp. 57-64, 1999

[Veig96] J.S. McVeigh, M.W. Siegel and A.G. Jordan, “Intermediate view synthesis considering occluded and ambiguously referenced image regions”, in *Signal Processing: Image Communication*, Vol. 9, pp. 21-28, 1996

[Wand95] B.A. Wandell, “Foundations of vision”, Sinauer Associates, Inc. Publishers, Sunderland, Massachusetts, 1995

[Weng92] J. Weng, P. Cohen and M. Herniou, “Camera calibration with distortion models and accuracy evaluation”, in *IEEE Transactions on PAMI*, Vol. 14, No. 10, pp. 965-980, 1992

[Wei94] G.Q. Wei and S.D. Ma, “Implicit and explicit camera calibration: theory and experiments”, *IEEE Transactions on PAMI*, Vol. 16, No. 5, pp. 469-480, 1994

[Wolt78] H.J. Woltring, “Simultaneous multiframe analytical calibration (SMAC) by recourse to oblique obervations of planar control distributions”, *SPIE Vol. 166, Applications of Human Biostereometrics*, pp. 124-135, 1978

[Woo96] W. Woo and A. Ortega, “Stereo image compression with disparity compensation using the MRF model”, *Proc. VCIP96*, SPIE Vol. 2727, No. 1, pp. 28-41, 1996

[Xu96] G. Xu and Z. Zhang, “Epipolar Geometry in Stereo, Motion and Object Recognition, A unified approach”, Kluwer Academic Publishers, The Netherlands, 1996

[Zane96] P.O. Zanen, "Single lens apparatus for three-dimensional imaging having focus-related convergence compensation", US Patent #5,532,777 July 2, 1996. Related patents US #5,828,913, October 27, 1998, and US #5,883,662, March 16, 1999

[Zcam] http://www.3dvsystems.com

[Zhan93] J. Zhang and J. Hanauer, "The mean field theory for image motion estimation", *Proc. ICASSP93*, Vol. 5, pp. 197-200, 1993

[Zhan96] Z. Zhang, "On the epipolar geometry between two images with lens distortion", *Proc. Int. Conf. Pattern Recognition* (*ICPR*), Vol. 1, pp. 407-411, 1996

[Ziss95] A. Zisserman, P.A. Beardsley and I.D. Reid, "Metric calibration of a stereo rig", in *Proc. IEEE Workshop on representation of visual scenes*, pp. 93-100, 1995

Summary

In the area of visual communication systems, much research effort is currently being put in the enhancement of quality and telepresence or immersiveness. Using very large displays or incorporating 3-D effects for example may cause enhanced telepresence. It is expected that this leads to communication systems that serve as alternatives for the conventional phone in interpersonal communication or for videoconferencing in order to reduce business travelling. Many other applications benefit from enhanced communication systems, in e.g. medical, industrial, entertainment and advertising areas.

In Chapter 1, we reviewed 3-D systems, ranging from the classic stereoscopic systems to the futuristic holographic video systems. The task of these systems is to introduce visual cues that are related to depth perception and not incorporated by conventional monoscopic displays. We identified three such cues. The first is the stereo cue, which means that the left and right eye of a viewer see a different image. This cue was already introduced long ago by the classic stereoscopic systems. The second cue is motion parallax. This means that whenever the viewer moves his head slightly, the images shown on the display will change also. Nearby objects appear to change more than objects that are far away, which is the additional depth cue. Further, the viewer can now ‘look around’ objects, providing the freedom to select a viewpoint of interest. The motion parallax cue requires adaptivity of the images shown, which has become possible only recently with the advance of digital signal processing. The third and last cue is the eye lens cue. This means that the eye lens actually focuses on the object that the viewer is looking at. All displays for video material do not serve this cue yet; regardless of the depth of the object one looks at, the eye lens focuses at the display plane. Holograms produce the eye lens, but the current state of technology does not allow for real-time holographic video systems with natural images.

We focused on adaptive multi-viewpoint systems. These provide a viewer with stereoscopic images, such as produced by a conventional stereo system, but also with motion parallax. We have considered a system on the basis of stereo equipment, that is, two cameras record the scene and a stereo display shows the scene. In between, a large amount of signal processing is needed to introduce viewpoint adaptivity. For the first time, such a system has recently been built in the European PANORAMA project, enabling real-time processing for a videoconferencing application.

We dealt with the signal-processing parts of the adaptive multi-viewpoint system as well as the integration of these components into the PANORAMA system. At the transmitter side of the system, stereo camera calibration and correspondence estimation are used to acquire a 3-D model of the scene. At the receiver side, viewpoint-adaptive stereo image synthesis is used for the visualization of the 3-D scene model.

In Chapter 2 we have extensively examined the modeling of stereo cameras. We introduced a new tensor-based notation that makes it easier to pinpoint the differences and similarities of the camera models currently available. Further we investigated lens distortion in great detail. The main reason to introduce it in camera models is to make the models more accurate. Besides this, we conjectured that it may also help to circumvent a theoretical proof that restricts the use of self-calibration.

In Chapter 3 we investigated stereo camera calibration algorithms. In this area, fixed calibration methods have come already to a level of maturity, while research is focusing on the more flexible but also more complex self-calibration methods. We proposed an algorithm for fixed calibration and derived a self-calibration algorithm from it, thereby unifying the two approaches. The algorithms have been designed in the Bayesian probability framework. Our experiments show that the fixed calibration provides model parameters that enable us to acquire a 3-D scene highly accurately. Our self-calibration algorithm deals with the most difficult case at hand: measuring all camera parameters without prior information, on the basis of only two images. In several experiments we found that the theoretical proof that limits self-calibration is not valid for cameras that include lens distortion, which opens the door to more generally applicable self-calibration algorithms. On the other hand, we found that the method is inherently unreliable. Now and then we found that the parameters were wrong compared to a ground truth with synthetic material. We could only observe this due to our approach with synthetic material. In other approaches that lack a ground truth false results may be obtained that appear accurate. A thorough investigation when the self-calibration approach including lens distortion works and when not, is an open research area. If solved, it may lead to reliable use of self-calibration methods, even in the most challenging case of single stereo image pairs as input material. Further, other existing self-calibration techniques must be evaluated also with synthetic data including the ground truth, to validate their true accuracy.

In Chapter 4 our goal was to derive a correspondence estimator that provides high quality and enables real-time implementations. We first reviewed many approaches to correspondence estimation (CE) and found that the Bayesian algorithms on the basis of Markov Random Fields (MRF) are the most promising regarding the quality. In the Bayesian framework, we found a new way to combine the hierarchical approach with the so-called cooling schedule in Simulated Annealing (SA) algorithms, in order to reduce the computation times by an order of magnitude. Using this framework, we derived two new algorithms: one for images from calibrated cameras, and one for uncalibrated stereo cameras. With the algorithm for calibrated cameras we obtained high quality 3-D models, judged by visual inspection. Further, the algorithm for uncalibrated cameras was able to cope with large differences in rotation and scale between the cameras. We made a major step towards real-time implementations with MRF algorithms, since the computational load of both algorithms was orders of magnitude lower than conventional algorithms due to the combination of the hierarchical framework and the SA algorithm. Many future research directions are still open. The robustness of the algorithms can still be improved, since about 5% of the runs did not converge to the right solution. Our algorithm still needs to be applied to a self-calibration algorithm. Further, Bayesian correspondence estimators are ideal algorithms to be extended, by e.g. incorporating temporal consistency constraints and other image-processing techniques such as image restoration. For these combined algorithms, the computational resources at this moment allow only implementations that perform the estimation on a one-by-one pixel basis instead of estimating all data simultaneously, which implies a causality constraint. Finally, a major area of research is the evaluation of our and similar algorithms with objective and meaningful quality measures. The measures we used were subjective, enabling rough evaluation but no quantitative comparison with other methods.

In Chapter 5 we derived an image synthesis algorithm that theoretically ensured geometrically correct visualization of 3-D scenes on stereo displays, able to be viewed from any position. In literature, many synthesis algorithms have been derived for the generation of virtual camera images, which is slightly different from our application and does not lead to correct scene visualization in general. Further, we examined in detail many sources of error that contribute to deviations in the geometry of the visualized scene. These include the effects of tracking errors, that is, errors in the measured viewer position, and rendering latency. These analyses are new for the area of multi-viewpoint systems. Several rules of thumb have been derived that enable a quick analysis, whether certain tracking errors or system latencies are visible or not. An extensive subjective test was performed that validated these theoretical findings. We also noticed that other means of interaction besides viewpoint adaptivity are needed in order to visualize 3-D scenes. These are (manual) means to manipulate the position, orientation and the scale of the scene, e.g. in order to display a very large scene on a normal sized display, or to see the rear side of a scene. An important result from our test is that both stereo systems and viewpoint-dependent systems were preferred over conventional monoscopic systems. The motion viewpoint dependency was said to enhance the visualization quality more than the stereo cue. Both statements argue for the incorporation of 3-D aspects in visual communication systems. There are several directions for future research. A major increase in viewer comfort can be established by a stereo display with better separation between left and right views. Further, although it was not found disturbing, the rendering latency of our system was clearly noticeable. This may be counteracted by means of the fast rendering techniques that are emerging in similar applications, like augmented reality systems.

In Chapter 6, we discussed the integration of all system components into the PANORAMA system. First we introduced a new image-based scene model that forms the heart of the PANORAMA system. It enables a real-time system implementation, and at the same time high quality scene models (in the order of 10^6 points). Then, we derived system settings in order to achieve a geometrically correct 3-D impression, while complying with all feasibility constraints of the real-time system. A constraint of more principal nature was identified for two-way videoconferencing applications with image interpolation algorithms: the recording cameras must be attached left and right of a display in order to provide direct eye contact, and thus the baseline is slightly larger than the display width. Together with the conditions for correct scene visualization, the stereo camera setup must be chosen carefully in order to have any overlap between the images and be able to acquire the scene in stereo. Extensive subjective tests have been conducted with the PANORAMA system. The main result of these tests is that the feeling of telepresence is greatly enhanced by the introduction of viewpoint adaptivity. The resulting motion parallax is a promising feature yielding a positive subjective impression. Even test persons with low expectations of 3-D techniques prior to our test, changed their opinion in favor of 3-D after their experience with the PANORAMA system. These tests showed that 3-D aspects do and will contribute substantially in visual communication systems, since they provide the viewer with more realistic and natural impressions.

In Chapter 7 we discussed the future of 3-D visual communication systems. The PANORAMA multi-viewpoint system was the first real-time system to be realized, and obviously, it can be improved both on conceptual and detail levels. More complex scenes may be captured by multiple camera systems or other modality devices such as range cameras. Reliable self-calibration algorithms are needed in order to have full freedom in the camera setup and the possibility to change it dynamically. The analysis of multiple images of more complex scenes requires improved image analysis techniques, e.g. better correspondence estimation algorithms that can deal with more than just head-shoulder scenes. We also need to use appropriate scene models, e.g. on the basis of light fields or wire frames. At the visualization side, real-time visualization algorithms can be implemented with unrestricted viewpoint-adaptivity and means for manual interaction. Further, it remains to be investigated to what extent the geometrical correctness of the visualized scene is necessary. The more the requirement can be relaxed, the more freedom we have in the setup of cameras and displays. The development of displays still has many possible directions. The telepresence feeling may be enhanced by higher resolution, larger displays (e.g. room size) and better stereo separation between left and right views. The eye lens accommodation cue still remains to be introduced. Further, the number of people that simultaneously can see the scene undistorted is still limited. This may be overcome by e.g. fixed multi-view displays that show so many views simultaneously that a complete audience experiences the stereo and motion parallax cues, without eye wear. In the application area, normal interpersonal communication is still mostly performed by phone. Ideally, the phone is replaced by the 3-D systems we examined, providing a wide-scale application. For actual videoconferencing, it is expected that multi-point communication is needed among much more than two people. Such applications are already emerging as research topics. The investigation of transmission of 3-D scene models using conventional video transmission channels is vital for the evolution of current monoscopic video systems towards 3-D systems on a global scale. In the more distant future, where the transmission standards may be redefined completely, holographic systems may serve as the ultimate 3-D visual communication systems. They solve many of the aforementioned limitations. At this moment, prototype holographic systems are available for real-time dynamical scenes. However, the synthesis part of such a system requires a massive parallel supercomputer, while only synthetic scenes can be shown on a display with the size of a human hand. Real-time acquisition of dynamical holograms is not yet possible. In the meantime, the gap may be bridged by hybrid systems that acquire scenes with normal cameras and visualize them with holographic displays.

Samenvatting

Multi-viewpoint systemen voor 3-D visuele communicatie

Op het gebied van systemen voor visuele communicatie wordt op dit moment veel onderzoek verricht. Het doel is hierbij het verbeteren van de algemene beeldkwaliteit en het verhogen van de zogenaamde tele-aanwezigheid, ofwel het gevoel werkelijk aanwezig te zijn bij de getoonde scène. Dit kan bijvoorbeeld bewerkstelligd worden met zeer grote beeldbuizen of met het aanbrengen van 3-D effecten in de getoonde beelden. Het is te verwachten dat zulke systemen aantrekkelijke alternatieven zijn voor de ons welbekende telefoon, of voor videoconferentie systemen om het zakenreizen te verminderen. Tal van andere applicaties heeft ook baat bij verbeterde videocommunicatie systemen, bv. voor medische, industriële, amusements en reclame doeleinden.

In hoofdstuk 1 namen we 3-D systemen onder de loep, uiteenlopend van de klassieke stereosystemen tot holografie. Het doel van deze systemen is om in beelden visuele kenmerken te introduceren, die gerelateerd zijn aan het waarnemen van diepte en die nog niet in gewone monoscopische videosystemen aanwezig zijn. We vonden drie van deze kenmerken. Het eerste is het stereoscopische kenmerk, wat inhoudt dat de kijker een ander beeld ziet met zijn linkeroog dan met zijn rechteroog. Lang geleden was dit al geïntroduceerd in de klassieke stereosystemen. Het tweede kenmerk is bewegings-parallax. Dit betekent dat als de kijker zijn hoofd beweegt, hij ook daadwerkelijk de scène vanuit een andere hoek waarneemt, waarbij objecten ten opzichte van elkaar een andere positie zullen innemen. De relatieve beweging van objecten is gerelateerd aan hun onderlinge diepte, hetgeen de kijker diepte-informatie verschaft. Dit kenmerk vereist dat de getoonde beelden voortdurend worden aangepast aan de kijkerpositie, wat pas sinds kort mogelijk is dankzij de vooruitgang in de digitale beeldbewerking. Het derde dieptekenmerk is gerelateerd aan het scherpstellen van de ooglens aan de diepte van het specifieke object waarnaar gekeken wordt. Alle huidige displays voor bewegende beelden zijn nog niet in staat dit kenmerk te genereren; de ooglens stelt altijd scherp op de beeldbuis, ongeacht naar welk object wordt gekeken. Hologrammen zijn wel in staat het ooglens kenmerk te genereren, maar deze kunnen nog geen bewegende beelden tonen van natuurlijke scènes.

Wij hebben zogenaamde multi-viewpoint systemen onderzocht. Deze verzorgen de stereoscopische en bewegings-parallax diepte-kenmerken. We keken naar systemen op basis van stereo apparatuur, dat wil zeggen, twee camera's nemen de scène op en een stereo display toont de scene. Ertussen zorgt een grote hoeveelheid beeldbewerking voor de bewegings-parallax. Zo'n systeem is recent voor het eerst gebouwd in het Europese PANORAMA project, resulterend in een 3-D videoconferentie systeem.

Wij hebben alle delen van de beeldbewerking onderzocht inclusief hun integratie in het PANORAMA systeem. Aan de zendkant worden stereo camera calibratie en correspondentie-schattingsalgoritmen gebruikt om een 3-D model van de scène te verkrijgen uit de opgenomen beelden. Aan de ontvangkant wordt een adaptief, stereo beeldsynthese algoritme gebruikt om het 3-D model te visualizeren voor de kijker.

In hoofdstuk 2 onderzochten we het modelleren van stereo camera's ten behoeve van calibratie. We introduceerden een tensor-gebaseerde notatie die het ons makkelijk maakt om verschillen en overeenkomsten aan te duiden in de grote hoeveelheid beschikbare cameramodellen. We hebben lens vervorming tot in detail bekeken, omdat we vermoedden dat dit naast een kwantitatieve verbetering ook kan leiden tot een meer flexibel gebruik van zelf-calibratie algoritmen.

In hoofdstuk 3 bekeken we camera calibratie algoritmen. Op dit gebied hebben de zogenaamde gefixeerde calibratie algoritmen al een status van volwassenheid bereikt, en concentreert het onderzoek zich op de meer flexibele maar ook meer complexe zelf-calibratie algoritmen. We hebben een algoritme ontworpen voor gefixeerde calibratie en leidden daaruit een algoritme af voor zelf-calibratie. Hiermee unificeerden we beide aanpakken. De algoritmes zijn volledig ontworpen binnen het Bayesiaanse raamwerk van de kansrekening. Onze experimenten tonen aan dat het gefixeerde calibratie algoritme camera-parameters oplevert die ons in staat stellen om 3-D scène modellen van hoge kwaliteit te verkrijgen. Ons zelf-calibratie algoritme ziet zich geplaatst voor de lastigste taak op het gebied van calibratie: het meten van alle parameters van een stereo camera, zonder enige voorkennis, en met alleen twee beelden voorhanden. In diverse experimenten vonden we ons vermoeden bevestigd dat het modelleren van lens vervorming resulteert in een meer algemeen bruikbaar en flexibeler algoritme voor zelf-calibratie. Aan de andere kant vonden we ook dat de methode inherent onbetrouwbaar is. Zo nu en dan waren de gevonden parameters volledig verkeerd. Dit laatste konden we alleen waarnemen door het gebruik van synthetisch beeldmateriaal waarbij de ideale parameters beschikbaar zijn. In andere aanpakken waar echt beeldmateriaal gebruikt wordt kunnen dus slechte parameters gevonden worden die goed lijken. Een braakliggend gebied voor onderzoek is de betrouwbaarheid van (bestaande) zelf-calibratie algoritmes.

Ons doel in hoofdstuk 4 was om een correspondentieschatter te ontwerpen die hoge kwaliteit combineert met snelheid. Eerst bekeken we verschillende aanpakken en vonden dat de Bayesiaanse algoritmes op basis van Markov Random Field (MRF) modellen de beste kandidaten zijn gezien hun hoge kwaliteit. We vonden een nieuwe mogelijkheid om snelheidswinst te boeken door de hiërarchische (HA) zoekmethode te combineren met de zogenaamde koelings-procedure in Simulated Annealing (SA) algoritmen. Met deze gereedschappen ontworpen we twee algoritmes: één voor al gecalibreerde camera's en één voor ongecalibreerde camera's. Met het eerste algoritme verkregen we 3-D scènemodellen van hoge kwaliteit, beoordeeld met visuele inspectie. Het algoritme voor ongecalibreerde camera's kon omgaan met zeer grote verschillen in rotatie en schaal (zoom) tussen de camera's van het stereo paar. Voor beide algoritmes geldt dat we een grote stap hebben gemaakt richting het gebruik van MRF-algoritmen in real-time toepassingen, gezien de snelheidswinst geboekt door de HA/SA combinatie. Vele richtingen zijn nog beschikbaar voor toekomstig onderzoek. De robuustheid van onze algoritmes kan verbeterd worden, want 5% van alle resultaten was niet goed geconvergeerd. Onze algoritmes dienen ook nog te worden toegepast in een zelf-calibratie algoritme. Verder zijn Bayesiaanse MRF algoritmes ideale kandidaten om verder te worden uitgebreid met bv. temporele consistentie restricties of gelijktijdige beeldrestauratie. Een groot gebied van onderzoek is het ontwikkelen van een objectieve maatstaf om de verkregen 3-D modellen te evalueren.

In hoofdstuk 5 hebben we een beeldsynthese algoritme ontworpen dat een geometrisch correcte visualizatie van 3-D modellen garandeert op stereo displays, waarbij de scène bekeken kan worden vanuit ieder willekeurige positie. In de literatuur zijn al vele soortgelijke algoritmen ontworpen voor het genereren van virtuele camerabeelden, maar deze verschillen licht van ons algoritme en bieden dan ook in het algemeen geen geometrisch correcte visualizatie. Verder bekeken we vele andere bronnen van fouten die kunen bijdragen in geometrische afwijkingen. Hierbij behoren fouten van de meetapparatuur voor de kijkerpositie en opgelopen vertragingen door de tijdsduur van het beeldsynthese algoritme. Deze analyses zijn nieuw op het gebied van multi-viewpoint systemen. We formuleerden diverse vuistregels om te zien wanneer de fouten leiden tot waarneembare geometrische afwijkingen. Een uitgebreid experiment met testpersonen valideerde onze theoretische bevindingen. Daarnaast vonden we ook dat nieuwe interactie-methoden gewenst zijn in multi-viewpoint systemen, zoals mogelijkheden om de getoonde scène manueel te verplaatsen, te roteren en te schalen. Een belangrijk resultaat was dat zowel stereo als multi-viewpoint systemen werden geprefereerd boven conventionele monoscopische systemen, waarbij het bewegings-parallax kenmerk meer zou toevoegen dan het stereo kenmerk. Dit is een sterk argument voor het introduceren van deze 3-D kenmerken in visuele communicatiemiddelen. Diverse richtingen staan nog open voor verder onderzoek. Het kijkers-comfort kan nog aanzienlijk toenemen als de scheiding tussen linker- en rechterbeeld van het stereo paar verbeterd wordt. Verder waren geometrische afwijkingen door de beeldsynthese-vertraging niet storend maar wel duidelijk zichtbaar. Dit kan verholpen worden door snelle renderingstechnieken toe te passen zoals al gebeurt in bijvoorbeeld 'augmented reality' systemen.

In hoofdstuk 6 integreerden we alle systeemdelen tot het PANORAMA 3-D videoconferentie systeem. Eerst introduceerden we een nieuw beeld-gebaseerd 3-D scène model, dat de grondslag vormde voor het PANORAMA systeem. Dit model maakt een real-time systeem mogelijk, waarbij ook de kwaliteit van de modellen hoog is (in de orde van 10^6 scène punten). Daarna leidden we de parameters af waaronder het systeem nog steeds een geometrisch correcte 3-D visualizatie kon opleveren, waarbij tegelijkertijd aan alle implementatierestricties werd voldaan. Daarnaast identificeerden we een restrictie die optreedt in 2-weg communicatiesystemen wanneer interpolatie gebruikt wordt voor de beeldsynthese, zoals in het PANORAMA systeem. De opnamecamera's moeten dan aan beide zendzijden links en rechts van het display gemonteerd worden, anders kan er geen direct oog-contact gemaakt worden. De afstand tussen de camera's is daarmee altijd iets groter dan de breedte van het display. Samen met de voorwaarden voor geometrisch correcte visualizatie moet de stereo setup van de camera's dan met zorg gekozen worden, anders overlappen de camerabeelden niet en kan de scène niet in stereo worden opgenomen. Uitgebreide experimenten met testpersonen zijn uitgevoerd met het PANORAMA systeem. Het belangrijkste resultaat hieruit is dat het tele-aanwezigheidsgevoel aanzienlijk wordt versterkt door de introductie van het bewegings-parallax kenmerk. Zelfs testpersonen met lage verwachtingen van 3-D technieken, veranderden hun mening na de ervaring met het PANORAMA systeem. Onze testen laten duidelijk zien dat 3-D kenmerken een grote en gewenste toevoeging zijn aan visuele communicatiesystemen, aangezien ze de kijker een meer realistisch en natuurlijk beeld verschaffen.

In hoofdstuk 7 bediscussiëerden we de toekomst van systemen voor 3-D visuele communicatie. Het PANORAMA multi-viewpoint syteem was het eerste gerealiseerde real-time systeem, en kan daarom uiteraard verbeterd worden, zowel op conceptueel niveau als in details. Complexere scènes kunnen opgenomen worden door systemen met meer dan twee camera's, of camera's met andere modaliteiten zoals afstand-camera's. Betrouwbare zelf-calibratie algoritmes zijn nodig om meer vrijheid te hebben in het kiezen van de camera setup en de mogelijkheid om deze dynamisch te veranderen. De analyse van meerdere beelden van een complexe scène vereist ook verbeterde beeldanalyse technieken, bijvoorbeeld correspondentieschatters die meer dan alleen hoofd-schouder scènes aankunnen. Ook het type 3-D scène model zal anders moeten worden gekozen, bijvoorbeeld lichtveld- of draadmodellen. Aan de visualizatiekant kunnen beeldsynthese algoritmen gebruikt worden met ongelimiteerde positievrijheid voor de kijker en mogelijkheden voor manuele interactie. Verder dient nog onderzocht te worden in welke mate de geometrische correctheid van de scène werkelijk nodig is. Hoe meer hiervan kan worden afgeweken, hoe meer vrijheid we hebben in het kiezen van de setup van de camera's en het display. De ontwikkeling van displays kan nog vele kanten op. Het tele-aanwezigheidsgevoel wordt vergroot door hogere resolutie, grotere displays en betere stereoscheiding. Het ooglens kenmerk dient nog steeds geïntroduceerd te worden. Verder, het aantal mensen dat de getoonde scène gelijktijdig zonder vervorming kan waarnemen is nog steeds gelimiteerd. Dit kan mogelijk verholpen worden door multi-view displays die vele beelden tegelijk laten zien, en zo verschillende kijkers tegelijkertijd de stereo en bewegings-parallax kenmerken verschaffen. Op het applicatiegebied zal idealiter de telefoon worden vervangen door een 3-D video systeem. Voor videoconferenties zal een multi-weg systeem nodig zijn in plaats van een 2-weg systeem, hetgeen nu al heeft geleid tot nieuwe onderzoeksprojecten. Het onderzoek naar transmissie van 3-D scene modellen via conventionele videokanalen is cruciaal voor de evolutie van de huidige monoscopische TV systemen naar een wereldwijd 3-D TV systeem. In de verder afgelegen toekomst, waar wellicht ook de transmissiesystemen volledig zullen worden vervangen, kunnen holografische systemen hun intrede doen als het ideale 3-D communicatiemiddel. Prototypes zijn op dit moment al gemaakt, maar voor het synthesegedeelte zijn zeer grote supercomputers nodig en voor het beeldanalyse deel is nog geen technologie. Dit gat kan wellicht worden gedicht met hybride systemen die scènes opnemen met gewone camera's en ze visualizeren met holografische displays.

Acknowledgements

It is only after writing a thesis, that you can truly enjoy a day of doing absolutely nothing. This thought occurred to me especially when I noticed that this thesis contains about half a million characters. Luckily I was in good company during this work.

First of all I would to thank my supervisor Emile and my professor Jan Biemond, who were my sounding board and helped in transforming vague thoughts into solid ideas. Emile, I enjoyed your famous yearly Spoorsingel drinks. You were a valuable room-mate, and maybe after all I even miss all those telephone calls that I had to answer while you were out…. Further I would like to thank all participants of the PANORAMA project, whose work and enthousiasm contributed to this thesis. I would like to thank especially Hans and Ad (and be carefull with the Bols).

I also would like to thank the members of the Mediamatics Lab, in specific Annet, Jeanine and Ben for their mental and undefinable support. Cees, we never agreed on some law of uncertainty but that is exactly what makes research valuable. The lunches with all of the young dogs were unforgettable. Erik, the piano will be finished this century, I promise, and please, never call me a spaghetti programmer again. Gerhard, the sun does shine (whenever you enter the room). Peter, I’d like to lunch with you again but only if one of us takes a tranquilizer.

I would like to express my gratitude to my new employer, Philips Research, who has been so kind as to provide me with a laptop computer to write this thesis during train travelling.

Also I would like to thank my friends, who did not forget me even when I was behind the computer most of the time. I thank especially the Leiden Club for their delicious meals, the ‘korst’ for their deviant behaviour, and the ‘klik’ for our Sunny-Ser meetings. Elke and Jan-Willem, thanks for being my paranymphs. Jan-Willem, thanks for designing the cover of the thesis, it is undeniably JW.

I would like to thank my family, especially my parents just for being there for me. Finally, Mar, or ‘Actie-Mar’, thanks for staying with me during all of those half a million characters and being such a huge source of energy.

Curriculum Vitae

André Redert was born in Vlaardingen, The Netherlands, on February 19, 1971. In 1989, he received his VWO diploma from the O.S.G. Prof. Casimir in Vlaardingen. In the same year, he started the study for electrical engineer at the Delft University of Technology. In 1993, he worked for three months as a trainee at the Rijks Universiteit Leiden on neural networks regulating human organs. In 1995, he received the M.Sc. degree from the Delft University of Technology. The master thesis work focused on the analysis of accuracy and resolution of wavelet transforms, used for the analysis of dolpin whistle sounds. After that, he worked for a short period at KPN Research in Leidschendam on quantitative analyses of telephony.

In 1995 he started his Ph.D. work at the Delft University of Technology in the Mediamatics Lab. (formerly the ICT Group). The work was embedded in the European ACTS PANORAMA project, sponsored by the European Union. This thesis provides a condensed overview of the work done on real-time 3-D visual communication systems. In 1998, he began with the restoration of an antique Viennese fortepiano. This major time-consuming operation is still continuing. At the beginning of the new millennium 2000, he started working on consumer 3-D television systems at Philips Research, Eindhoven.