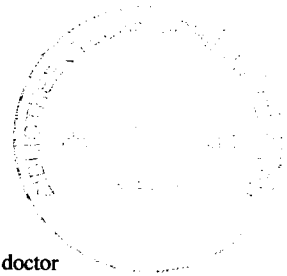
Manfred Ziegler

REGION-BASED ANALYSIS AND CODING OF STEREOSCOPIC VIDEO

AKADEMISCHER VERLAG MÜNCHEN · 1997

Region-Based Analysis and Coding of Stereoscopic Video

PROEFSCHRIFT



ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.dr.ir. J. Blaauwendraad
in het openbaar te verdedigen ten overstaan van een commissie,
door het College van Dekanen aangewezen,
op maandag 13 oktober 1997 te 10:30 uur

door

Manfred ZIEGLER

Diplom Informatiker, Technische Universität München

geboren te München, Duitsland

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. J. Biemond

Toegevoegd promotor:
Dr.ir. R.L. Lagendijk

Samenstelling promotiecommissie:

Rector Magnificus	Technische Universiteit Delft, voorzitter
Prof.dr.ir J. Biemond	Technische Universiteit Delft, promotor
Dr.ir. R.L. Lagendijk	Technische Universiteit Delft, toegevoegd promotor
Prof.dr. M. Strintzis	Aristotelian University of Thessaloniki
Prof.ir. G. Honderd	Technische Universiteit Delft
Prof.dr.ir. E. Backer	Technische Universiteit Delft
Prof.dr.ir. F.C.A. Groen	Universiteit van Amsterdam
Dr. E. Hundt	Siemens AG, Munich

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Ziegler, Manfred

Region-Based Analysis and Coding of Stereoscopic Video

.-München: Akad. Verl., 1997

Zugl.: Delft, TU, Diss., 1997

ISBN 3-929115-96-4

Gesamtherstellung: dm druckmedien, München, Tel: 089/2802099

Table of Contents

Summary

Humans use binocular vision to judge depth to see the world "three-dimensionally". Binocular vision is the vision achieved with two eyes, exploiting the difference between the images in the left and right eye. With computer vision, two cameras a certain distance apart "replace" human eyes, and so simulate the natural system. Such stereoscopic systems will become more and more important in the context of telepresence for future telecommunication applications.

As the available bandwidth for transmitting video signals is limited, it will be necessary to reduce the video data rate while maintaining an acceptable video quality. Video data rate reduction is achieved by using image data compression techniques which are an attempt to minimise the large bandwidth requirements and so reduce the costs of transmission but still provide acceptable image reconstruction. When dealing with stereoscopic images, the original data rate is doubled, as there are two images instead of one to be transmitted, which makes compression even more important. Image acquisition with two cameras also causes a displacement between the left and right spatial image in a stereoscopic image pair. This displacement, referred to as disparity, is a unique phenomenon associated with stereoscopic images. Since it is inversely proportional to depth, it can be used to analyse stereoscopic image pairs. In this thesis, a new *stereoscopic image sequence coder* which makes use of disparity is developed. Most of the work leading to this thesis was for the European project DISTIMA (DIgital STereoscopic IMaging & Applications).

Over the last couple of years, object-based coding, a new coding concept, has attracted a great deal of attention. By transmitting the shape, motion and colour of the objects in an image, it is possible to avoid the annoying coding errors, such as mosquito effects and blocking artefacts, produced by block-oriented, hybrid coding. Furthermore, important image areas such as facial details in face-to-face communications can be reconstructed with a higher image quality than with block-oriented hybrid coding. Moving objects are the main problem. Whenever two separate objects move towards each other, no information is available in object-based coders to indicate which object is covering the other. This will cause a large error if the wrong object is chosen when the image is reconstructed. With the help of stereoscopic information, it is possible to overcome the limitations of this kind of coding scheme. Using depth information, which can be estimated from the stereoscopic signal, it is easy to decide which object is farther away and which is visible. For this reason, it is desirable not only to transmit the three parameters shape, colour and motion, but also a fourth parameter - depth or disparity. Disparity can be used to determine the position of an object in space. As the coder developed in this thesis

will not be used for real physical objects, but on image regions which do not necessarily correspond to physical objects, the coder is referred to as a *region-based* coder.

This thesis describes a *region-based stereoscopic image sequence coder* which is based on the principles of image analysis and image synthesis. The source model uses rigid and arbitrarily shaped regions undergoing translational motion instead of fixed blocks of pixels. The regions are found by segmenting the images by means of motion and disparity vectors. These regions are then described by a set of parameters including colour, shape, motion and disparity, which are extracted by the image analysis step. To reduce the bit rate, the parameters are subsequently coded using standard coding schemes such as temporal and spatial prediction and entropy encoding. Image synthesis then uses these parameters to synthesise the temporally next image and the corresponding spatial image in an stereoscopic image pair. The major developments in this thesis include:

- A pixel-accurate *disparity and motion estimation* . This is required to segment regions of this kind. Also developed in this thesis is a disparity estimator of this kind which does not need any knowledge of the stereoscopic geometry and so can handle practically any kind of stereoscopic image pairs. It is based on a dynamic programming approach and takes the feature difference of the pixels into account, assuming piecewise-smooth, inner disparity regions, as well as the relationship between disparity jumps and occlusions in a stereoscopic image pair. The resulting vector fields are pixel-accurate and of high quality, as the analysis shows. Therefore, they are a good basis for further processing, say, the segmentation of regions according to their disparity. The experiments in this thesis which also apply dynamic programming to motion estimation show that the estimation of a motion vector field is also possible with this approach. With the region-based coder, it can, therefore, be used not only to estimate disparity but also to estimate motion.

- *Image analysis* , based on these vectors and the current left image. First of all, initial regions are segmented in accordance with the source model. Segmentation is based only on the disparity and motion vector fields. After a suitable number of regions are obtained by merging, the four required parameters - disparity, motion, shape and colour -are extracted for each of the remaining regions. Evaluation shows the high quality of the segmentation of the image into regions and the subsequent description of the regions. As the regions are only changed if new regions can be merged with them in a subsequent image, temporal consistency of the regions can also be guaranteed in a sequence.

- *Image synthesis*, performed on an image using the parameters which are stored in a region-memory. Image synthesis, in principle, is a straightforward process involving the reconstruction of the regions from the transmitted parameters and putting them at the correct position in the image. This is done by motion compensating the regions in the left image sequence and disparity compensation in the right images. Evaluation shows that image

synthesis delivers a synthesised image with high visual quality. The region memory makes it possible to build up a database of regions in the image sequence, so decreasing the required transmission bit rate if, say, a previously visible region is covered and then becomes visible again.

As the main coder requirement is reducing the number of bits to be transmitted, all the parameters have to be coded efficiently. Different *coding strategies* are adopted depending on the nature of the parameters. Motion, disparity and shape parameters undergo lossless coding using spatial and temporal prediction schemes. The colour parameters are coded using shape-adaptive DCT. This makes it possible to describe arbitrarily shaped regions efficiently.

Whenever uncovered background or occlusions occur, or if image analysis fails to describe the scene completely, say because of imperfections in the source model, there will be a synthesis error in the synthesised image. Despite this synthesis error, an *informal subjective evaluation* shows that the visual quality of an individual stereoscopic image pair is comparable with that of an MPEG2-encoded video signal, although the latter requires a considerably higher bit rate than the region-based coder developed in this thesis.

When the quality of the entire encoded sequence without synthesis error addition is assessed, several inconsistencies in the temporal behaviour of the region-based coder can be identified. These inconsistencies produce noticeable artefacts which will be corrected by adding the synthesis error. These artefacts can be caused by the incorrect merging of regions, the temporal jerkiness of regions due to incorrect motion vectors, and the sudden changes of the region colour due to a missing update of the region parameters at some time. To increase the quality of the encoded stereoscopic sequence, the synthesis error is added to the synthesised images. This will no longer be necessary when the region parameters are regularly updated.

Even with rudimentary synthesis-error coding - simple vector quantisation when the error is above a threshold - the region-based coder achieves a similar subjective quality as an MPEG2 coder when the two stereoscopic channels of a sequence are encoded separately with the same total bit rate.

The investigations in this thesis only address the use of rigid, two-dimensional regions in translational motion and it is still an open question whether the efficiency of the parameter coding can be increased by using a different source model, say flexible two-dimensional or three-dimensional regions. The coding efficiency of the synthesis error can be increased by using a more intelligent error coding which also takes the regions into account. Apart from algorithmic questions, further work will also include investigations into the possible applications of region-based stereoscopic coders in the near future. The European project PANORAMA (PAckage for New OpeRational Autostereoscopic Multiview systems and Applications) will be the vehicle for most of this research.

List of Symbols

A_i	Number of pixels in region $\Re_i$
ACV_m	Accumulated Cost Value
b	Camera distance (Baseline)
$B_{i,j}$	Number of common pixels on the boundary of neighbouring regions $\Re_i$ and $\Re_j$
c	Normalisation constant in *NFD* and *GNFD*
C	Colour information
C_i	Colour parameter for region $\Re_i$
C_l, C_r	Left / Right Camera
$CF_{n,m}(\vec{D})$	Cost Function
$\vec{D}$	Disparity vector $\vec{D} = \begin{pmatrix} D_h \\ D_v \end{pmatrix}$
D_h	Horizontal component of disparity vector $\vec{D}$ (= disparity)
D_v	Vertical component of disparity vector $\vec{D}$
$\vec{D}_i$	Disparity vector for region $\Re_i$
$\vec{D}_{n,m}$	Disparity vector for point *(n,m)* in the image plane
δ	Disparity Jump
δ_h	Horizontal disparity jump
δ_v	Vertical disparity jump
f	Camera focal length
$f(\delta)$	Function to punish a disparity jump δ
$FD_{n,m}$	Feature Difference for point *(n,m)*

$GACV_m$	Generalised accumulated cost value
$GCF_{n,m}(\bar{D})$	Generalised Cost Function
$Gf(\delta_h, \delta_v)$	Generalised $f(\delta)$
$GFD_{n,m}$	Generalised Feature Difference
$GNFD_{n,m}$	Generalised Normalised Feature Difference
i, j, k, l	Integer counters
$\bar{I}_i$	Mean luminance value of region $\Re_i$
L_i, R_i	i-th left / right original image
L'_i, R'_i	i-th left / right reconstructed image
$\bar{M}$	Motion vector $\bar{M} = \begin{pmatrix} M_h \\ M_v \end{pmatrix}$
M_h	Horizontal component of motion vector $\bar{M}$
M_v	Vertical component of motion vector $\bar{M}$
$\bar{M}_i$	Motion vector for region $\Re_i$
$\bar{M}_{n,m}$	Motion vector for point (n,m) in the image plane
$\bar{M}(n, m, k)$	Motion vector field from image $k-1$ to image k
$\bar{M}_{RM}(n, m, k)$	Motion vector field from the region memory to image k
(n, m)	Spatially discrete coordinates (= pixels)
(n, m, k)	Horizontal / vertical coordinates in image k
N	Number of pixels in the matching window $[(2 \cdot \tau + 1) \cdot (2 \cdot v + 1)]$
$NFD_{n,m}$	Normalised Feature Difference for point (n,m)
O_l, O_r	Occlusion in the left / right image
$P'_l(x'_l, y'_l)$	Projection of point P to the left image plane
$P'_r(x'_r, y'_r)$	Projection of point P to the right image plane
$P(x_p, y_p, z_p)$	Point P with world-coordinates

P_{DP}	Number of possible paths through the cost matrix in Dynamic Programming
PCV	Percentage of Correct Vectors [%]
PCS	Percentage of Correct Segmented pixels [%]
$PSNR$	Peak Signal to Noise Ratio
(π, λ)	Parameters for horizontal and vertical size of image
$QS_{i,j}$	Quality criterion for merging of small region $\Re_i$ to larger neighbouring region $\Re_j$
$QE_{i,j}$	Quality criterion for merging of equivalent regions $\Re_i$ and $\Re_j$
R_C	Bitrate required to transmit the colour
R_D	Bitrate required to transmit the disparity
R_M	Bitrate required to transmit the motion
R_S	Bitrate required to transmit the shape
$\Re_i$	Region with region number i
σ_i^2	Variance of luminance values of region $\Re_i$
σ_L^2, σ_R^2	Variance of luminance values in the left / right image
S	Shape information
S_i	Shape parameter for region $\Re_i$
(τ, ν)	Parameters for horizontal and vertical size of matching window
$\Theta_d, \Theta_h, \Theta_v$	Threshold values of disparity, horizontal and vertical motion, used in segmentation
$W_L(n,m),\ W_R(n,m)$	Luminance value at point (n,m) of left / right image
$\overline{W_L}, \overline{W_R}$	Mean luminance value of left / right image
(x, y, z)	3-D coordinates
ω	Number of regions

1. Introduction

Within seconds he ran out to the deck and waved and grinned at over three billion people. The three billion people weren't actually there, but they watched his every gesture through the eyes of a small robot tri-D camera which hovered obsequiously in the air nearby. The antics of the President always made amazingly popular tri-D: that's what they were for.

Douglas Adams, The Hitchhikers Guide through the Galaxy, 1978

In the next ten to fifteen years, emerging multimedia services will have a strong impact on social and cultural life. By the year 2010 the boundaries between computing, communications and broadcasting will have largely been eliminated. User-friendly multimedia terminals with flat panel displays then will provide access to a wide range of entertainment, communication, information and education services. Digital systems will allow the better use of existing infrastructures for TV distribution and will also improve image quality and definition (HDTV and 3-DTV). New digital systems will make it possible to increase the number of programmes and the number of sound channels for multi-lingual programmes. These new systems will also allow the creation of advanced interactive audio-visual services.

The standard television concept can be extended in future image communication systems to cover stereoscopic multiview, 3-D and full-space imaging. This will provide the user with potentially variable, controlled and three-dimensional windows of the world. As this new concept will give the viewer a feeling of "being present in the scene", it is called telepresence. Depending on the developments in display technology, various degrees of resolution and spatial perception will be offered. There is a large potential in these new types of imaging in non-broadcast applications (e.g. teleconferencing, medicine) and in the consumer entertainment market. Telepresence extends the video conferencing concept so that participants can use non-verbal aspects of communication (eye contact, spatial perception, body movement, gestures, facial expressions) in the same way as they would in a face-to-face meeting.

The degree of telepresence that can be achieved depends on the accuracy of sensory information transmitted to the user. New sensors and devices - for instance high resolution stereoscopic displays, navigational aids (head position/orientation tracking, eye-tracking, body tracking), data gloves as well as highly agile platforms for mobility in a remote environment - will create the need for a much higher bandwidth than is usual today.

One of the most important and challenging tasks facing telepresence is the transmission and presentation of visual information in a form that human beings are used to: namely three-dimensional. Looking at our daily life, one can see that the acquisition of visual information is a highly active process. It involves frequent changes in gaze direction, accommodation, convergence angle and a number of involuntary control mechanisms which serve to compensate for various limitations of the eye. Present video communications systems can only support a few of these activities, since pictures on 2-D displays do not provide the relevant information. A second obstacle is the large amount of data needed to be transmitted for these new telepresence systems.

On the display side, practical solutions for 3-D displays are available but they rely on special glasses and are therefore not applicable in interpersonal communications. Currently the most promising approaches to auto-stereoscopic multiview displays (without special glasses) use lenticular screen plates for optically addressing the viewer's left and right eye. Viewpoint-adaptive display techniques are being developed which use head-tracking devices to sense the actual viewing position of individual viewers and display the appropriate stereo views accordingly.

On the transmission side, advanced compression is required because of the large amount of data needed to be transmitted for telepresence systems. In the last decade a group called MPEG (Moving Picture Expert Group) has worked on compression standards called MPEG1 and MPEG2. Both standards are block-based coding schemes which subdivide an image into separate blocks and work on these independently. Recent results in the RACE DISTIMA project (DIgital STereoscopic IMaging and Applications) [Zie92a] have shown that it is possible to transmit stereoscopic signals - which consist of two spatially separated signals - compatible to MPEG2. However, such block-based coding schemes suffer from blocking artefacts, especially at a low bit rate. For that reason object-based coding, which is expected to have a better quality, is an upcoming topic in MPEG4 and other committees.

The transmission of video signals via computer and telecommunication networks will become more and more important. But depending on the used network, the available bandwidth is restricted. Applications via a telephone line only have a few kbit per second available, via ATM networks the possible data rate increases to a couple of Mbit per second, which is still not enough when dealing with TV-resolution images with an original data rate of 166 Mbit/s. Another aspect will be the billing of such future services: the more data one sends, the more one has to pay. Because of this it will be necessary to reduce the video data rate, which is much higher than data rates for data and speech transmission. Nevertheless the video quality still has to be acceptable. This video data rate reduction is achieved by applying image data compression techniques, which seek to minimise the large bandwidth requirements and hence reduce the cost of transmission with acceptable image reconstruction results. When dealing with stereoscopic

images, the original data rate is doubled, as there are two images instead of one to be transmitted, which makes compression even more important. Although stereoscopic features are exploited, a stereoscopic coding method will usually be based on already existing algorithms.

Most of these technologies are quite generic in nature and do not exclusively relate to telecommunications, nevertheless their successful development is likely to help in the introduction of advanced telepresence services.

1.1 Object-Based Coding

Within the last couple of years, a new coding concept - called object-based coding - has gained world wide attention: through the transmission of the object shapes, the two types of annoying coding errors of block-oriented hybrid coding - known as mosquito effects and blocking artefacts - can be avoided. Furthermore important image areas such as facial details in face-to-face communications can be reconstructed with a higher image quality than with block-oriented hybrid coding. As these object-based coding methods are based on an analysis of the images at the encoder and a synthesis of the objects at the decoder, they also are called object-based analysis-synthesis coding.

The main advantage of object-based analysis-synthesis coding methods compared to standard block-based methods is due to the fact that the object description avoids certain problems of fixed blocks. Whenever fixed blocks are used, these blocks do not define an object boundary accurately. The fact that a block can actually belong to two neighbouring objects is a source of large errors, as the two objects might move differently but only one displacement vector will be assigned to all of the block. Further the emerging synthesis error will be quantised based on this block. Therefore coding errors such as blocking artefacts and so-called mosquito effects can be observed in the decoded images when a block-based scheme is used. Object-based coding helps to overcome these disadvantages, as all the calculations are based on objects, so the above-mentioned problems will not occur if the object description is accurate.

The main disadvantage of object-based coders is that present implementations are known that can only handle very restricted sequences. All these coders aim at the very low bit-rate coding of videophone sequences in particular, where only a few objects are moving and some knowledge about the content is available. The main problem is with moving objects, as no information is available in such object-based coders to indicate which object is covering another one. With the help of stereoscopic information it is possible to overcome the restrictions of this kind of coding scheme. Using the depth information it can be decided easily which object is farther away and which one is visible. Through such stereoscopic extensions of the existing

object-based coding concepts not only videophone scenes can be handled, but also general scenes. Dealing with a stereoscopic signal is a further step towards telepresence. All this leads to the concept of an object- or region-based stereoscopic coder.

1.2 Region-Based Stereoscopic Coding

An object-based coder transmits the parameters shape, colour and motion. With this information the next image in time will be synthesised. Current implementations of object-based coders are restricted to scenes where only a few moving objects are shown. The problem is that in general scenes objects can be occluded by other objects. This will always happen if there are two objects with contradictory motion. With the help of disparity information it can be decided which object is in front. Now the objects can be ordered according to their distance to the camera and information given on which object is visible. Because of this additional information more robust and flexible coding approaches can be designed. For this reason it is desirable not only to transmit the three parameters shape, colour and motion, but also a fourth parameter: disparity or depth. As the coder developed in this thesis will not work on real physical objects, but on image regions which do not necessarily correspond to physical objects, the coder is called *region-based.*

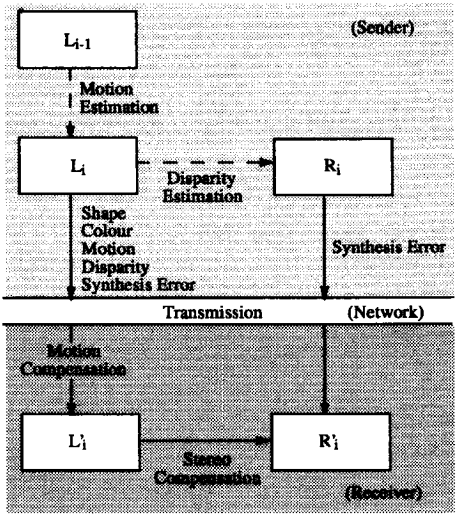


Figure 1.1: Principle of the region-based stereoscopic coder

In the region-based stereoscopic coder described in this thesis and shown in Figure 1.1, regions will be segmented according to their motion and disparity. These regions will be motion-compensated in one of the two image sequences of the stereoscopic signal and stereo-compensated from this synthesised image in the other. If the disparity is used in addition to the parameters of usual object-based coders, the receiver gets all the information necessary for the synthesis of both images. This way no additional information has to be transmitted to synthesise the second channel of a stereoscopic system. The additional overhead to be transmitted will then be restricted to a possible synthesis error signal.

The coding scheme presented in this thesis uses disparity information as an additional piece of information, and as a result does not need any a priori knowledge about the content of the scene.

1.3 Outline of the Thesis

This thesis is based on techniques of coding schemes known as object-based analysis-synthesis coding. When setting up a region-based stereoscopic coder, a knowledge of stereoscopic principles is required. Chapter 2 gives a brief survey of these stereoscopic principles and the coding techniques used in this thesis. This includes an explanation of the basic geometrical rules of a stereoscopic system, the definitions of disparity and occlusions, as well as an overview of the principles and current implementations of object-based analysis-synthesis coders and stereoscopic coders. An essential part of a stereoscopic coder is the estimation of the disparity vectors.

In Chapter 3 an improved disparity estimation algorithm based on dynamic programming is developed and discussed. This includes an explanation of the dynamic programming as well as its application to find an optimal solution to the problem of disparity estimation. In several steps the original concept of dynamic programming is extended until the final system is able to estimate highly accurate disparity vectors to be used for the segmentation of regions. In a region-based coder this segmentation of regions and their handling is the most critical point.

Chapter 4 describes the necessary image analysis tools - such as initial segmentation, merging of regions and the extraction of the regions' parameters - and their adjustments to the needs of this system. Also the image synthesis which synthesises the image at the receiver site is described here. Different synthesis methods are investigated in order to get the best possible synthesis.

The complete system utilizing many known components from image coding algorithms is described in Chapter 5. This includes the coding of the parameters extracted by image analysis, the handling of the synthesis error left after image synthesis and the network aspects of the developed coder, as well as the results of a statistical and an informal subjective evaluation comparing the region-based stereoscopic coder to a standard MPEG2 coder.

Finally, the achieved results and topics for future research in order to improve the performance of the region-based stereoscopic coder will be discussed in Chapter 6.

2. A Survey on Stereoscopic Image Coding

The development of the region-based stereoscopic coder in this thesis is based on techniques used in object-based analysis-synthesis coding. Present implementations of these coding techniques aim at the efficient coding of scenes, where
- only a few objects are moving,
- object motion is dominant and moderate,
- the moving objects cover up to 40-60% of the image area and
- no camera motion occurs.

Based on these assumptions [Höt92], investigations in object-based analysis-synthesis coding have up to now been restricted to typical videophone and videoconference applications aiming at a very low bit rate transmission [CCL95, Mus95, MVD96, PS94]. New approaches as described in the MPEG4 Verification Model [MPEG96] also aim at the coding of general scenes, but a closer look shows they are actually still based on blocks.

Through the use of stereoscopic information it will be possible to overcome the above mentioned restrictions of object-based coders. A short overview of the two basic ingredients of the coding scheme, the stereoscopic and the two-dimensional image coding principles is given next. Furthermore, a selection of the most common stereoscopic image coding schemes will be presented and discussed.

2.1 Stereoscopic Principles

Humans use binocular vision to judge depth to see the world "three-dimensionally". Binocular vision is the vision achieved with two eyes, exploiting the difference between the images in the left and right eye. While binocular vision only provides one of many depth clues, it is the one that seems easiest to understand. In computer vision, two cameras at a certain distance apart will "replace" the human eyes, and thus simulate the natural system setup. Points on the surfaces of objects are depicted in different relative positions depending on their distances from the viewer [Hor86]. The key to an automated stereo system therefore is a method for determining which point in one image corresponds to a given point in the other image.

A short overview of the basic geometrical principles is given next, followed by the definition of *disparity* and a discussion on its estimation.

2.1.1 Basic Geometry

It is important to understand the geometrical principles of stereoscopic imaging. These principles will be used throughout this thesis, either as a basis for image processing or to provide simple solutions to the problems described later on. In stereoscopic imaging, two cameras have to shoot two spatially separated images. These cameras can be set up in a way that the optical axes are parallel or are inclined to each other by a certain angle, called the convergence angle. If the optical axes are parallel, a lot of problems can be solved quite easily. Unfortunately in most cases the optical axes are inclined which exacerbates the problems. However, if the camera parameters such as focal length and convergence angle are known, it is possible to make use of the geometry and simplify the way solutions are found.

Figure 2.1 shows the simplified setup of a stereoscopic system, assuming that the two optical axes of the two cameras C_l (left) and C_r (right) are parallel and separated by a distance b. The line connecting the lens centres is called the *baseline*. This baseline is perpendicular to the optical axes and parallel to the horizontal axes (x-axes) of the images.

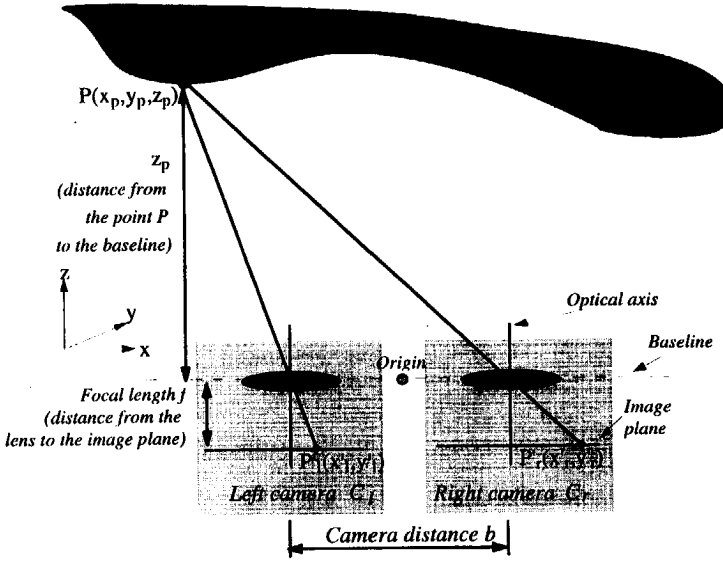


Figure 2.1: Stereoscopic system setup with parallel optical axes

The coordinates of a point $P(x_p, y_p, z_p)$ in the "real world" are measured relative to an origin midway between the lens centres. They can be calculated from the image coordinates - which are measured relative to the centre of the lenses - $P'_l(x'_l, y'_l)$ in the left and $P'_r(x'_r, y'_r)$ in the right image if f - the distance from the lens centre to the image plane in both cameras - is known. Then the following equations can be set up:

$$\frac{x'_l}{f} = \frac{x_p + b/2}{z_p} \tag{2.1}$$

$$\frac{x'_r}{f} = \frac{x_p - b/2}{z_p} \tag{2.2}$$

$$\frac{y'_l}{f} = \frac{y'_r}{f} = \frac{y_p}{z_p} \tag{2.3}$$

From equations (2.1) and (2.2) it follows that

$$\frac{x'_l - x'_r}{f} = \frac{b}{z_p} \tag{2.4}$$

The difference in image coordinates $x'_l - x'_r$ in equation (2.4) is called *disparity*. By solving (2.1) - (2.3) the real-world coordinates of the point P can be obtained:

$$x_p = b\frac{(x'_l + x'_r)/2}{x'_l - x'_r} \tag{2.5}$$

$$y_p = b\frac{(y'_l + y'_r)/2}{x'_l - x'_r} \tag{2.6}$$

$$z_p = b\frac{f}{x'_l - x'_r} \tag{2.7}$$

As can be seen from equations (2.5) and (2.6) the disparity $(x'_l - x'_r)$ is linearly proportional to the distance between the lens centres b. Even more important in the context of stereoscopic imaging is that the distance of the object to the camera z_p is inversely proportional to the disparity, as can be seen from equation (2.7). Equation (2.7) also shows that the range of possible disparity values in a parallel setup is $]0, \infty]$, which means that a disparity of 0 never can occur.

A point in the environment visible from both cameras causes a pair of image points - one in the left image, the other one in the right image - called a *conjugate pair*. A point in the right

image corresponding to a specified point in the left image must lie somewhere on a particular line because the two have the same y-coordinate. This line is called an *epipolar line.*

A feature in the left image may or may not have a counterpart in the right image. If it does not have a counterpart, this is because this feature can only be seen in the left image but not in the right one. The feature is then called *occluded.* If the feature does have a counterpart, it must appear on the corresponding epipolar line. For this simple geometry all epipolar lines are parallel to the x-axis (Figure 2.2).

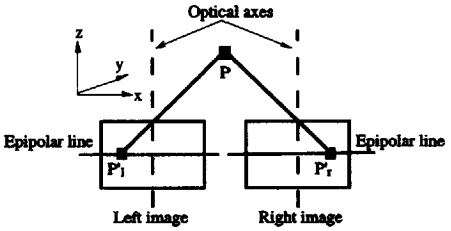


Figure 2.2: Epipolar lines for setup with parallel optical axes

Similar equations to (2.1) - (2.7) can also be setup for the more general non-parallel geometry. The convergence angle plays an important role in this case, making the equations a lot more complex. As the properties of the non-parallel case such as the relationship of disparity and depth are the same as for the parallel case, these equations will not be shown here. However, they can be found for instance in [Hor86]. The main difference is the range of possible disparity vectors. With a non-parallel set-up the disparity vectors can take values within $[-\infty, \infty]$ as will be shown later in Figure 2.6.

In this general case whenever the cameras are not aligned in parallel the epipolar lines will also not be parallel, neither to the x-axis nor to themselves (Figure 2.3). However, if the camera parameters are known, these epipolar lines can be calculated and therefore be used to solve some problems of stereoscopic imaging.

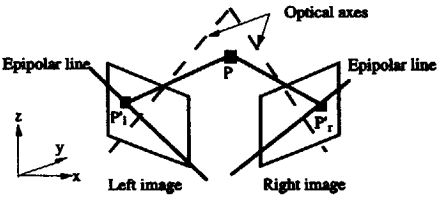


Figure 2.3: Epipolar lines for setup with non-parallel optical axes

2.1.2 Disparity

Every point in the left image of a stereo pair should have a corresponding duplicate (conjugate pair) in the right image positioned on an epipolar line. The point will be positioned in the right image with a displacement compared to its counterpart in the left image. The same holds for displacements from the right to the left image. The vector describing this displacement is called the *disparity vector* $\vec{D}$. The horizontal component D_h of $\vec{D}$ is called *disparity* in this thesis and its size depends on the distance of the object from the camera system (depth), as shown in Section 2.1.1. Consequently the disparity can help to determine the position of an object in space. The relationship between the disparity vector $\vec{D}$ and the disparity D_h is different for stereoscopic setups with the optical axes parallel and non-parallel axes. Figure 2.4 shows the case of parallel optical axes, whereas Figure 2.5 shows non-parallel axes. Point P_l in the left image, appears as point P_r in the right image. Due to the parallel geometry the epipolar lines are horizontal, therefore the two points are on the same horizontal line. The vector $\vec{D}$ pointing from point P_l to point P_r is also horizontal and it is identical to the disparity D_h because its y-component is zero.

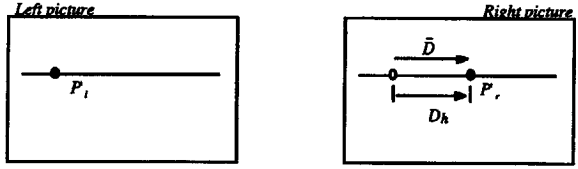
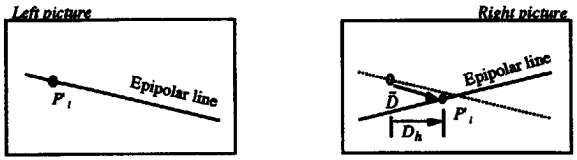


Figure 2.4: Definition of disparity in a setup with parallel optical axes

In the case of a setup with non-parallel optical axes, the disparity vector $\vec{D}$ will have both x- and y- components non-zero, as depicted in Figure 2.5. In theory the disparity runs along the epipolar line. It is approximately equal to the x-component D_h of the vector $\vec{D}$ under the assumption that the convergence angle of the cameras is not too large. In the test sequences used in this thesis [DIS92, DIS94] the convergence angle is less than 4 degrees, which leads to a sufficiently small y-component of the disparity vector of maximum 2 pixels [DIS94]. In an environment where one image is directly predicted using the disparity vectors and the other image, it would not be advisable to neglect this y-component. However, with the system described in this thesis a vertical component of maximum 2 pixels can be omitted, as can be seen in later Sections. Nevertheless the disparity vector has to be estimated in both dimensions, otherwise large vector estimation errors - also for the horizontal component - would result. Therefore reference to disparity implies reference to the x-component D_h of vector $\vec{D}$, as depicted in Figure 2.5. Likewise, reference to the epipolar line also implies a

reference to the line along which the x-component of vector $\vec{D}$ lies. With the sequences used this is almost the horizontal scanline of the image.



o : *Location of P't in the right picture*
D_h: *Disparity*
$\vec{D}$: *Disparity vector*

Figure 2.5: Definition of disparity in a setup with non-parallel optical axes

In this thesis the disparity value D_h gives the number of pixels a point in the left image has to be shifted in relation to the right image. As the general case is a non-parallel camera geometry, the disparity D_h (from left to right) can have the following values (see Figure. 2.6):

$D_h = 0$ if the object lies on the Vieth-Müller-Circle (VMC), indicated with the dashed line in Figure 2.6,

$D_h < 0$ if the point is inside the VMC (such as the black circle in Figure 2.6) and

$D_h > 0$ if the point is ouside the VMC (such as the black square in Figure 2.6).

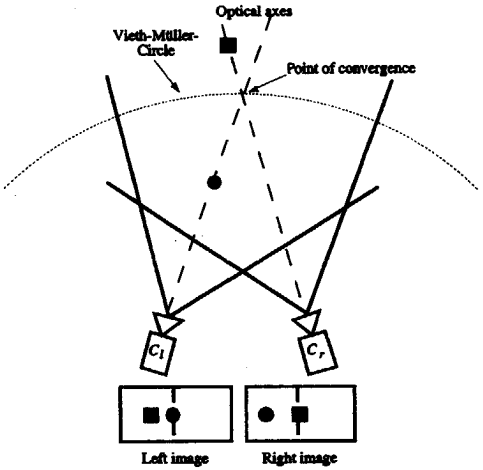


Figure 2.6: Disparity and object distance in a setup with non-parallel optical axes (not to scale)

2.1.3 Occlusions

A unique phenomenon in stereoscopic imaging is occlusion. In a stereoscopic image pair, there will be areas which are not visible in the left or the right stereo image. These areas are called occlusions.

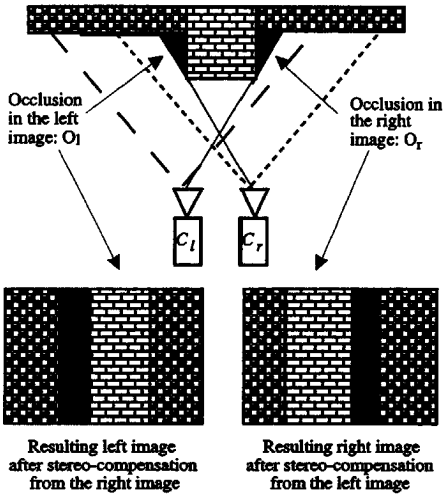


Figure 2.7: Occlusions in stereoscopic imaging

For points in the area O_l no corresponding image point can be found in the right image. The same holds for points in the area O_r as far as the left image is concerned. A method to estimate the disparity in a stereoscopic image pair (see Section 2.1.4) should therefore not assign incorrect vectors, but it should detect and mark these areas as occlusions. This should be possible independent of the way of matching - either left-to-right or right-to-left. Stereo-compensation shifts the pixels of the image according to the disparity vectors, this way predicting the other image and thus predicts the other image of the stereoscopic pair. The result of stereo-compensation will be an area of no information where occlusions occur (see Figure 2.7). Some simple rules can be set up:

- Occlusion will only occur when there are two neighbouring objects having a different distance to the camera (different disparity values).
- In the left image of the stereoscopic pair this occlusion will always occur left of an object which is closer to the camera than a neighbouring object. In the right image it is the other way around.

In [Liu95] the visibility of any impairments in occluded areas in stereo vision is investigated. The result was that these impairments are as visible as errors in regions that are present in both images, the so-called binocular regions. Therefore, for the purpose of stereo image coding, irrelevant data elimination in occluded areas has to follow the same rules as in binocular regions.

2.1.4 Disparity Estimation

Disparity estimation is the key problem in stereo imaging. In applications such as depth estimation for modelling or prediction of one stereoscopic image from the other one, the first task to be solved is the determination of corresponding points in the two stereoscopic images. The *correspondence problem* is that of identifying features in two images that are projections of the same entity in the three-dimensional world. Most of the methods applied for disparity estimation are well-known from motion estimation, where the correspondence of two points has to be found in two temporally apart images. However, there are two major differences in disparity estimation compared to motion estimation:

- In disparity estimation the possible range of vectors is extremely large compared to motion estimation. This leads to a higher computational effort, and - as there are a lot more possibilities to investigate - also to less reliable results. Therefore disparity estimation is a more difficult task than motion estimation.

- A point on the surface of an object might not be visible from either camera - this way introducing an occlusion - but if it does appear in both images, then the two image points must lie on corresponding epipolar lines as shown above.

On the other hand, using the knowledge of the epipolar geometry, the search area in which to look for the corresponding point in the other image, can be reduced to a one-dimensional search along this epipolar line. In the general non-parallel camera setup these epipolar lines are not identical to the horizontal lines of an image. By calculating the equation of an epipolar line [PD96] - which requires knowledge about the camera parameters and the camera geometry - the disparity estimation can be done searching only along the points of the epipolar line to determine the best match. Another approach is to transform one of the images (using again the camera parameters) such that the epipolar lines will be parallel to each other and re-sample it. This process is called rectification [PD96]. The epipolar geometry is then no longer needed for disparity estimation itself because the epipolar lines are now parallel. Additionally corresponding lines now even have the same y-coordinate as they are also parallel to the horizontal line of the images. As the stereoscopic coding scheme in this thesis is intended to be applicable to general scenes, neither calibrated cameras nor any knowledge about the epipolar lines can be assumed. A calculation of the epipolar lines or a rectification

will therefore not be possible, and the search for the corresponding point in the other image has to be two-dimensional along the x- and y-axes.

The most common way to find correspondences is by using a *correlation method* [BDP95, BN96]. Given a patch of one image the correlation with all patches in the search area of the other image is calculated. This method is well-known in image coding for the estimation of motion, but can also be used for estimation of disparity in the case of stereoscopic images. Normally these patches are square blocks. The size of these blocks greatly influences the result. If the blocks are too small, the brightness pattern will not be distinctive enough and many false matches may be found. Also noise will be a problem if the patches are too small. If they are too large, resolution is degraded, since neighbouring image regions with different disparities will be combined in the measurement. This might lead to a situation where the two blocks will not match, unless disparity is constant. As in motion estimation a multiple resolution scheme would appear to be the answer. First correlation matches on reduced images are found, then they are used to confine the search for matches in the next higher resolution pair of images [ANG95].

While correlation methods are often the first ones to be proposed, they do not perform very well for disparity estimation. Perhaps the most serious shortcoming is their sensitivity to differences in foreshortening [Hor86]: if a surface is tilted relative to the baseline, its projection will appear shorter in one image than in the other. This is shown in Figure 2.8, where the line $\overline{A'_l B'_l}$ in the left image is longer than $\overline{A'_r B'_r}$ in the right image, although both are projections of the same line $\overline{AB}$ in the 3-D space.

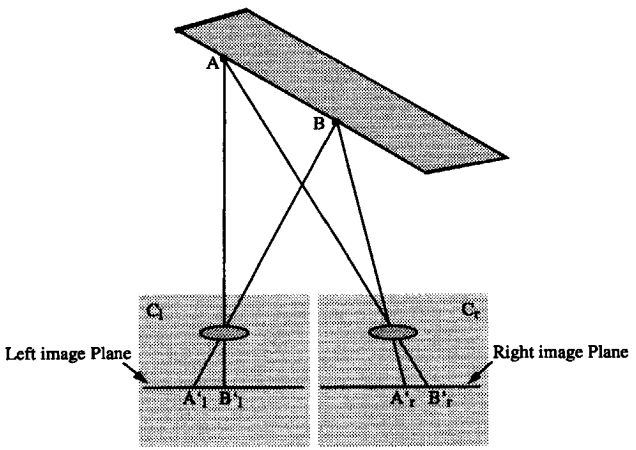


Figure 2.8: An inclined area with different amounts of foreshortening in two images

The two grey-level waveforms will not be well correlated in this case, which leads to wrong or rather approximate disparity values. A modification of the correlation approach that leads to usable results is the incorporation of warping in the images to undo the foreshortening effect. Also based on correlation, but first estimating a couple of candidate matches through the phase difference in the Fourier transformed signal, the *phase correlation* [Wal91] reduces the influence of the pattern size. Another solution is the use of dynamic programming to optimise the output according to a cost function [PZC95].

If the grey level is more or less constant in a part of the image, it is hard to find corresponding points. Correlation methods using patches smaller than the uniform region will produce no clear optimum. It might be more reasonable to estimate disparity only where there are rapid fluctuations, as on edges between patches of more or less uniform brightness or texture. These lead to the concept of *feature-based* or *edge-matching methods* [POA94, SSJ95]. In particularly simple cases, all edges visible in one image are also visible in the other image. Most systems do not work well if this condition is not satisfied, but even if they work well the outcome is a sparse disparity information which has to be interpolated. In [KD94] such a system combining feature extraction, matching and interpolation is proposed.

Another approach, *joint motion and disparity estimation* was recently proposed in different papers [TGS96, IE94]. In these approaches motion and disparity estimation is based on block matching algorithms, but taking into account the inherent coherence relation between disparity and motion. The resulting disparity values look very promising, although the computational effort is a lot higher than with the earlier mentioned algorithms. The main advantage of this method, i.e. a temporally consistent vector field, is very important for several stereoscopic applications, but it would not be exploited in the scheme developed in Chapter 4 of this thesis.

No matter which approach is used, disparity either will be estimated from the left to the right image or vice versa. Depending on which direction is chosen, this will have different effects on the detection of occluded areas (see Figure 2.7). Although different areas in the image will be classified as occlusion, all the methods are applicable for both left-to-right and right-to-left disparity estimation without any changes of parameters.

2.2 Two-dimensional Image Coding

In image coding the main objective is to reduce the data rate. There are two basic possibilities: redundancy reduction and irrelevance reduction.

Redundancy reduction is an information-preserving coding step. The parts of the data which can be predicted through their statistical properties will be removed. An example of redundancy reduction is the well-known Huffman Code. It is used to assign code words to the source symbols to minimise the average bit rate. The code assignment in Huffman coding depends on the long term probability of occurrence of the source symbol. The more often a source symbol occurs - and the more probable it is - the shorter will be the code word assigned to it. Longer code words will be assigned to improbable source symbols that do not often occur. Decoding of the code words can always reconstruct the data without any difference to the original code words.

The goal of irrelevance reduction is to adjust the coding scheme to the maximum capacity of the human visual system: parts of the information that the human observer is not able to see will be removed. To reduce irrelevance in image coding, quantisation of the transmitted values is carried out. This is a non-reversible step, meaning that some information is lost. Irrelevance reduction therefore leads to differences between original and decoded image, which is described as the quantisation error.

2.2.1 Source Models and Coding Principles

Most of the well-known image coding schemes are combinations of redundancy and irrelevance reduction. A very important point to be considered in constructing a scheme for image coding is the choice of the source model [Mus95] to be used. The source model influences the parameters to be computed and transmitted and so has a high impact on the coding efficiency. One of the main criteria for the decision of what source model to use is whether a single image or image sequences should be coded and transmitted. Table 2.1 shows a list of the most common source models.

In this table and the following description, colour refers to a Y, U, V format of the images, thus including luminance (Y signal) as well as chrominance information (U, V signals). This colour information is coded using a transformation scheme such as the Discrete Cosine Transformation (DCT), yielding the transformation coefficients.

Source model	Coding principle	Parameters to be transmitted
Pixels	Predictive coding	Prediction error
Blocks of pixels or total image	Transform coding	Transformation coefficients
Translational moving blocks of pixels	Hybrid transform coding	Motion vectors, transformation coefficients of the prediction error
Moving unknown objects	Object-based analysis-synthesis coding	Motion, shape, colour
Moving known objects	Knowledge-based coding	Motion, shape changes, colour changes
Facial expressions	Semantic coding	Motion, shape, colour action units

Table 2.1: Influence of the source model on the parameters to be transmitted [Mus95]

In *predictive coding* an attempt is made to predict the pixel to be encoded. The prediction is made using the encoded values of the previously transmitted (and encoded) pixels. Only the prediction error (differential signal) is used for transmission. In adaptive predictive coding, the prediction can be based on local image statistics or by varying the coarseness of the quantiser. These variations can be based on visual criteria or by not transmitting the prediction error whenever it is below a certain threshold (as in conditional replenishment). A further possibility is delayed coding where the encoding of a pixel is delayed until the "future trend" [NH88] of the signal can be observed and then coded to take advantage of this trend.

In *transform coding* [Cla85], blocks of pixels or the whole image are transformed into blocks of data, called coefficients. This is done in order to reduce the correlation of the signal samples by changing the statistical properties of the data. Again the irrelevance is reduced by quantising the coefficients and transmitting only a certain number of coefficients. Cosine transforms have become most popular because they are well matched to the statistics of the image signal. An adaptive transform coding scheme can be made by changing the transformation such that it matches the image statistics or by changing the criteria for

selection of the coefficients in order to match the subjective quality requirements. Well-known examples of this kind of coding are the JPEG standard and its derivatives [PM93, RH96, Lin95]. Other possibilities include wavelet [OS95, Sha93, QCH94] and subband coding [Woo91, LP96]. Figure 2.9 shows a simplified block diagram of such a hybrid transform coder. The actual parameters to be coded are the motion vectors and the colour information of a block. In a local feedback loop - which is an embedded decoder - the image will be reconstructed and stored in the image memory to be available for the differential coding of the next image in time.

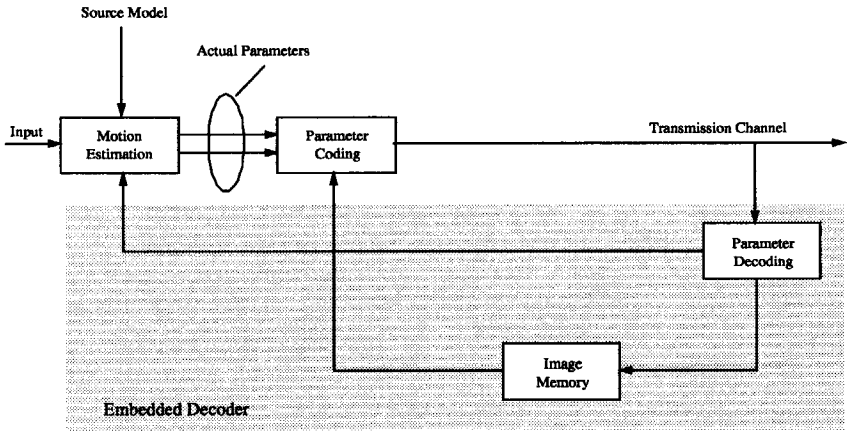


Figure 2.9: Simplified block diagram of an hybrid transform coder

Predictive coding and transform coding have been designed for the coding of single images. For the coding of image sequences also the temporal correlation of two successive images will be taken into account. In *hybrid transform coding*, this temporal correlation is reduced by applying a motion compensation of blocks of pixels. The motion information has to be transmitted in this case. This is done as a single "motion vector" per block. In addition a linear transformation of the residual image is followed in order to reduce the spatial correlation. This principle is used in the image compression standards H.261 and H.263 [RH96, GSF95], where a DCT is combined with motion compensation.

System approaches to *object-based analysis-synthesis coding* have been based on 2D and 3D moving objects as the source models for segmentation. Each moving object is described and encoded by three parameter sets defining its motion, shape and colour which are analysed at the sender site. At the receiver site the image can be synthesised from these parameters fairly easily. Results up to now have been shown using head-and-shoulder videophone images,

which identify an easy-to-model class of images, suitable for high compression coding [MHO89].

In cases where it is mainly a certain known moving object which has to be encoded - e.g. a moving human head - *knowledge-based coding* can be applied. Here the coding efficiency can be increased by using an explicit object model such as a predefined wireframe model, whose parameters can be adapted to the shape of the real head [AHS89]. The predefined wireframe model allows a better fit of the shape by fewer vertices.

In *semantic coding* an even higher abstraction level is used. Facial expressions like the opening of an eye or mouth, the most common example, can be described by one or several so-called action units as described in [EF77]. A temporal change of the action unit is associated with defined changes of several vertices of the 3-D model. Thus the change of one action unit can be encoded instead of the changes of several vertices in order to increase the coding efficiency.

International standardisation committees like ITU-T (International Telecommunication Union - Telecommunication) and ISO (International Standard Organisation) have established a number of standards for image coding. H.261 and H.263 [RH96, GSF95] are video coding standards released by ITU-T which combine motion compensation with a Discrete Cosine Transformation (DCT). The MPEG (Moving Pictures Experts Group) committee (formed by ISO-IEC/JTC1/SC29/WG11) produced the MPEG1 (1.15 Mbit/s video and audio) and MPEG2 image compression standards [RH96, Mat95]. The main emphasis of all these standards is a high compression factor. They all use translational moving blocks as the source model, in order to allow a simple hardware structure of the coder. The MPEG group is about to establish a new standard MPEG4 [CJB94]. In addition to the compression issue, a number of new functionalities, e.g. object-based access to the data, are introduced, which opens up the possibility of using a source model having either known or unknown objects as the basis. The resulting coding principle will be explained in more detail in the next Section.

2.2.2 Object-Based Analysis-Synthesis Coding

In contrast to block-based image coding, object-based analysis-synthesis coding [MHO89] allows arbitrarily shaped objects to be described by means of motion, shape and colour parameters. This requires the additional transmission of the shape parameters compared to block-based schemes. These parameters depend on the choice of the objects. Common objects to be used in object-based analysis-synthesis coding include 2-D and 3-D objects, either rigid or flexible.

Object-based analysis-synthesis coding subdivides each image of a sequence into objects and describes each object i by three sets of parameters defining Motion $\vec{M}_i$, Shape S_i and Colour C_i of the object. Figure 2.10 shows the concept and structure of an object-based analysis-synthesis coding scheme.

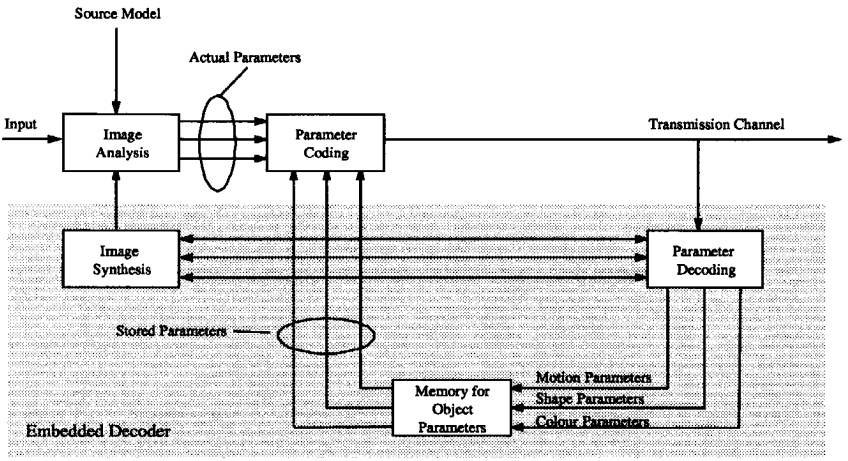


Figure 2.10: Block diagram of an object-based analysis-synthesis coding scheme [MHO89]

Instead of the image memory of hybrid transform coding techniques (see Figure 2.9), object-based coding requires a memory to store the parameter sets $\vec{M} = \{\vec{M}_i\}$, $S = \{S_i\}$ and $C = \{C_i\}$ of the objects. The object memory of the coder and the decoder contain the same parameter information and allows the encoder and the decoder to reconstruct a transmitted image by image synthesis. As the encoder makes use of the temporal correlation of successive images, the embedded decoder allows the use of a decoded image for image analysis of the next input image at the coder. However, the analysis fails in image areas which cannot be described by the source model being applied. These areas will be treated separately as "special objects", the so-called MF (Model Failure) objects.

The parameters $\vec{M}_i$, S_i and C_i of each object i have to be coded efficiently. In [MHO89] the motion parameters $\vec{M}_i$ and the shape parameters S_i are coded using predictive coding methods. For the prediction the information from the previous image is used. In the case of planar rigid objects the shape information describes the silhouette of an object. Therefore contour coding schemes [SSG96, TW96] are applied and only the temporal changes of the silhouette are encoded. The colour information is normally encoded by hybrid coding techniques, which combine motion compensated prediction with transform-coding of the residual image [Höt92].

While block-oriented hybrid coding techniques transmit only two parameter sets (the motion and colour information of each block), object-based analysis-synthesis coding transmits three parameter sets (the motion, shape and colour information of each object). Therefore, the additional bit rate R_S, required for transmitting the shape information S has to be compensated by a reduction of the bit rates R_M and R_C required for motion $\vec{M}$ and colour information C, in order to achieve at least the same coding gain. This can be done in two ways. First, only one motion parameter is transmitted for a complete object, or secondly the prediction of the colour information can be improved through the use of a more advanced and appropriate source model, which could also allow object rotations in addition to translations.

The efficiency of object-based coding largely depends on the size of the objects. The larger the objects, the higher the coding efficiency. Therefore, in the case of small objects, most object-based analysis-synthesis schemes use a block-based coding scheme as a fall-back solution. This way it can be guaranteed that the performance of the coder is always superior or at least equal to that of block-oriented coding.

2.2.3 Different Implementations

Object-based analysis-synthesis coding has been proposed for instance in [MHO89, Ost90, AHS89, Koc91, NHC91, GK87, Höt90, TAB95]. All these implementations encode arbitrarily shaped regions instead of square blocks. However, there are still three main characteristics, which differ:

- the source model (either 2-D objects or 3-D objects),
- the model of motion (either translational motion only or affine motion) and
- the colour coding strategy.

Apart from the specific implementation, one of the essential problems of object-based analysis-synthesis coding is the image analysis part, especially the subdivision of a scene into objects. All object-based coding schemes up to now use the motion information for this task, assuming that all parts of one object will move with the same velocity in the same direction. Several approaches are known to detect moving objects and to measure their velocity using either translational or affine motion. Many of them are based on the evaluation of the optical flow [Adi85, Pot75, Ull79]. Based on this approach the interdependence between motion and the object boundary is not taken into account. In [MHO89] therefore a joint motion estimation and object boundary detection is proposed.

Another important question is how to encode the colour information. Some systems, for example the MPEG4 Verification Model [MPEG96] use block-based schemes, even when dealing with arbitrarily shaped regions. To do so, blocks not covered 100% by the object have to be filled with some estimated colour from the object. This way quite a lot of information has to be coded, which then will not be used for the reconstruction. Another approach is

fractal coding [FLB94] or Vector Quantisation [DK95]. Again the underlying structure is still block-based, in this case using a quadtree decomposition to describe the object. It seems logical to go one step further and use colour coding methods with non-block objects.

One way of coding the colour information with non-block objects is to approximate the YUV-values with polynomial functions [Koc83, Leo87], called polynomial approximation. However the problem with polynomial approximation is that only "smooth" areas can be described properly, whereas the extension to higher polynomial functions to describe "rough" areas is difficult. For that reason polynomial approximation might not be able to achieve the required quality. A second solution is region oriented transformation coding [GEM89]. Here the question is how the appropriate orthogonal functions can be found for the regions. Using, for example, the orthogonalization scheme of Schmidt or Householder, this is a computational expensive procedure. Extrapolation [KA93, Kau95] aims at a shape-independent description using the circumscribing rectangular. To do this a computational expensive regularisation is necessary. A final possibility is the shape-adaptive DCT [SM95]. Although the 2-D correlations are not completely exploited using a shape-adaptive DCT, it is preferable for current realisations, as the predefined orthogonal set of DCT basis functions can be used, which makes the algorithm easy and fast [SBM95, Sik96]

2.3 Stereoscopic Image Sequence Coding

When dealing with stereoscopic image sequences, one has to handle two image sequences instead of one as described in the previous Sections. The most straightforward method to encode a stereoscopic image sequence would be to encode the two sequences separately, each with one of the two-dimensional coding schemes described before. Such a method does not evaluate the spatial correspondence of the stereoscopic image pair. This results in either a higher data rate or a lower quality because the spatial information is not exploited [ZT92].

In stereo-compensated coding, one image sequence (either the left or the right one) still has to be coded with a conventional two-dimensional coding scheme, whereas the second image sequence can be predicted using disparity information. In this Section existing implementations of stereo-compensating coders - based on either hybrid transform coding methods or object-based analysis-synthesis coding - will be discussed. Special attention will be paid to the role of disparity and the effects of occlusions.

2.3.1 Implementations of Stereoscopic Image Sequence Coders

Most of the current implementations of stereosopic image sequence coders are further developments of existing hybrid transform coders. Within the European project DISTIMA [Zie92a] a coding principle based on MPEG2 was investigated. In this system, the left image is coded in accordance with the MPEG2 standard. This system was hardware implemented within this project and successfully demonstrated at the end of 1994 [Hor93, SV93]. The developed prototype is shown in Figure 2.11. For the compression of the right image, it is possible to choose either motion-compensated or stereo-compensated coding per block, depending on what gives the better quality (see Figure 2.12).

Figure 2.11: Prototype of the DISTIMA coder

The DISTIMA project not only was the first one to make use of the spatial scalability concept in MPEG2, it still is the only fully MPEG2-compatible stereoscopic system built and demonstrated in hardware. Later approaches to stereo coding as [YC94] have been based on these results, reducing the encoding effort through applying motion and disparity estimation on subsampled images. Other MPEG2-compatible systems are described in [Cha95, PKH95, TA95, TA94]. They determine ways in which temporal scalability concepts can be applied to exploit redundancies between the two views of a stereoscopic scene.

Another approach - which is not based on a standard coding system - linking 2D left and right motion was suggested by [CDP95]. This is done in order to define a coding scheme using a unique channel for motion estimation and compensation. It is a two step algorithm applying a dynamic monocular analysis and a static binocular analysis afterwards. This way the 3-D motion and structure parameters are determined and can be used for motion compensation. Although the system performs motion compensation for both the stereoscopic images, disparity information is used only for the analysis of the parameters.

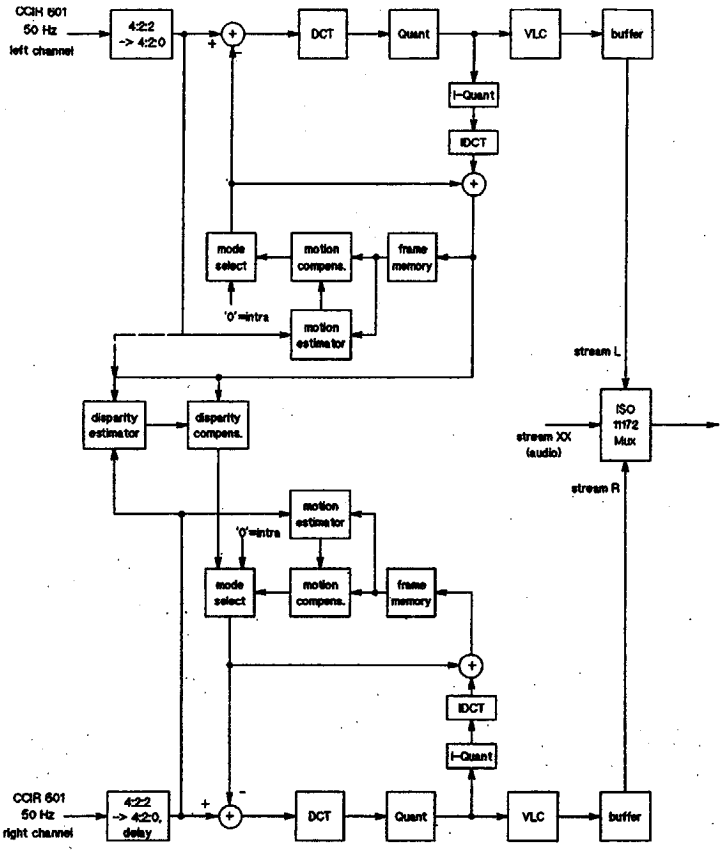


Figure 2.12: Blockdiagram of the DISTIMA coder

As in conventional 2-D image coding, the trend in stereoscopic image coding is also towards object- and region-based systems. First proposals for an object-based stereoscopic coder were published in 1993 [ZP93]. This system segments image regions based on disparity and motion information and uses these regions for the prediction of the next image in time or the alternative view of the stereoscopic signal. For the coding of the prediction error block-based methods have been used. Later on this approach was improved, for instance, by improving the disparity estimation [PZC95] or using more exact error-coding methods [ZP96]. This thesis describes and evaluates a complete system including all the elements proposed in [PZC95] and [ZP96].

Various other approaches to improve individual parts of the system have been recently published. In [FLB94a], for example, the vector field estimation and object segmentation was investigated. A 6 parameter affine model as in [AW93] is used, which is capable of describing translation, zoom, rotation and scaling. This affine displacement model can be extended to true 3-D, which contains information about the orientation of the plane in which the affine displacement occurs. The estimation of the parameters of this model is not performed directly. Initially a conventional translational vector field estimator is used. The vector field which is found is then partitioned into blocks of vectors and with the aid of linear regression initial affine parameters are found for the blocks. Next these affine parameters are clustered and remapped to the image in order to provide a segmentation. This segmentation is then used in the refinement of the affine parameters. The clustering/affine parameter refinement process is repeated iteratively until the result converges.

In [TGS95] a split and merge segmentation procedure based on 3-D motion and disparity is proposed. The information obtained is used to determine regions with similar motion and depth parameters. This is combined with a depth modelling method that offers full depth information at the decoder site. The segmentation part of the algorithm is interleaved with the estimation part in order to optimise the coding performance of the segmentation procedure. Motion and depth model parameters are then quantized and transmitted to the decoder along with the segmentation information. An object-based motion-compensating scheme is used to reconstruct the images based on the objects created by the segmentation approach.

2.3.2 Disparity and Occlusions in Stereo Coding

In a stereoscopic image sequence coder - no matter whether block-based or object-based - occlusions are a source of a large error when stereo-compensation is applied. In occluded areas there is no information available to be used for the prediction of the alternate image. These areas will have to be transmitted as residual error after applying a coding method. In stereoscopic image sequence coding, this led to an approach where one channel was predicted using motion information - by having a coder compatible with the chosen standard two-dimensional coder - and the second channel was predicted either motion- or stereo-compensated, depending on what method resulted in the better prediction [Hor93]. In experiments the motion-compensation mode was chosen for image areas which either were occluded or had very large estimated disparity vectors. In both cases a disparity estimation was not possible, either due to occlusions or as the search area for the disparity vectors was too small due to the hardware requirements of the coder. On the other hand, stereo-compensation showed its advantages in areas with fast motion, where motion estimation reached its limits.

2.4 Conclusion

In this Chapter a survey on the stereoscopic principles, two-dimensional image coding and stereoscopic image coding was given. Based on the geometrical rules of stereoscopic imaging and the "classical" methods for two-dimensional image sequence coding several implementations of stereosopic coders were discussed. What all of them have in common is the exploitation of the disparity information. The disparity vectors can be used to predict one of the images of the stereo pair, similar to the motion compensation known from the two-dimensional coding. Therefore disparity estimation is one of the key problems in the realization of a stereoscopic coder. Due to the extremely large range of possible disparity vectors this is not an easy task. However, with the knowledge of the epipolar lines the estimation can be restricted to a one dimensional search along these lines. Even without knowledge about the epipolar lines the search area can at least be reduced. Also occlusions have to be handled effectively. As they follow simple rules they can even be used to improve the quality of a disparity field. With an object-based analysis-synthesis coder this information will not be used for fixed blocks, but for a compensation of objects. Therefore the segmentation of the images into objects or regions is a crucial part of such a system. Up to now no region-based analysis-synthesis stereoscopic coder has been presented in the literature. This thesis points to a system of this type, which will be described in the following Chapters.

3. Disparity Estimation

Disparity is a unique phenomenon associated with stereoscopic images or stereoscopic image sequences. It describes the displacement between the spatially left and right image in a stereoscopic image pair. In stereoscopic image coding, disparity can be used to spatially predict one image of a stereo pair from the other, so reducing the number of bits to be transmitted. Disparity will be the key to segmenting an image into regions and shifting these regions according to their disparity in order to predict the second image. Disparity has to be estimated from a pair of stereoscopic images. The principles used for estimating disparity as well as for estimating motion are based on a comparison between two images with a temporal (motion) or a spatial (disparity) difference. Both methods aim to find the correspondence between image points in the two images. The main difference is that with disparity estimation there is implicit information about which part of the object is occluded. This is because there is a relationship between disparity and depth, but there is no such relationship for motion. This Chapter discusses an approach to implementing disparity estimation which uses dynamic programming and takes this knowledge into account. The estimation is based on an evaluation of the luminance values in the images, but the result will be used for the chrominance signal as well.

3.1 Rationale and System Overview

The two most common methods to solve the correspondence problem are block-matching methods and feature-based methods:

- Block-matching methods attempt to solve the correspondence problem by comparing a block from one image to blocks at possible matching positions in the other.
- Feature-based methods match special features - either single points, edge sections or whole edges - to find the correspondences.

With both methods, the result is a vector which associates either a block of pixels or a feature to its best match. For the entire image, the outputs are two displacement-vector maps, one showing the displacement in x-direction, the other in y-direction.

In the case of disparity estimation, there is a one-to-one correspondence between the epipolar lines in a stereo pair of images. When the epipolar geometry is known, disparity estimation can be restricted to a one-dimensional search along the epipolar lines. As this geometry is unknown in many applications, it would be useful to be able to estimate disparity without this knowledge.

A general approach of that kind is developed in this Chapter: The epipolar geometry will be ignored and a y-displacement added to the search. Although this is computationally more expensive, it makes the system flexible because the parameters of the cameras are not required to calculate the epipolar geometry and the algorithm is capable of handling a wide range of image pairs.

In the region-based stereoscopic coder described in this thesis, the displacement-vector maps are used to segment the image regions which are then encoded and transmitted to the receiver. Segmentation by means of vector maps has been shown to be quite robust [Kir89]. However, the following requirements have to be fulfilled:

- *Generality*: To be able to estimate disparity in the general case - without any information about the cameras - the method used for disparity estimation in this thesis also searches in the y-direction.

- *Density*: To have a good base for segmentation, the vector field has to be dense, which means one value per pixel is needed. For pixels without a displacement vector, it cannot be decided to which region they belong.

- *Smoothness*: As each region has to be described in terms of its contour and colour, the larger the objects are, the better it is for coding. To get large regions with the segmentation process later on, the vector field has to be smooth, which means random fluctuations in the vector field have to be minimised.

- *Accuracy*: As there will be a clustering of pixels with the same displacement vector, these values have to be close to the "real" values. Otherwise there will be a large error after region-compensation in the coder. Additional bits for coding the stereoscopic synthesis error signal would then be required.

Existing methods - such as the MPEG2 Video Simulation Model [MPEG90] - use block-based correlation approaches to perform the necessary motion or disparity estimation. However, even though these methods address the requirements referred to above, the results are inadequate, as they estimate only one vector per block. This does not give the necessary density and accuracy per pixel. Feature-based approaches deliver more realistic values, but produce only sparse vector fields which have to be interpolated afterwards to obtain a dense

vector field. As with block-based methods, the necessary accuracy per pixel cannot be achieved.

A new disparity estimator will be presented in this Chapter. It can calculate a dense and smooth disparity vector field, containing well-estimated disparity values that can be used subsequently for segmentation. The new disparity estimation system is shown in Figure 3.1. The inputs to the system are the left L_i and right R_i images. Due to slightly different camera characteristics, the left and right images may have a luminance discrepancy. Therefore, before the images are used for further processing, some preprocessing based on statistical difference modelling is required to adjust the two images (*Correction of Luminance Difference* block). In order to reduce the amount of computation, only some of the theoretically possible disparity values will be investigated. A second preprocessing step, the search range, is used to limit the range of disparity values that are investigated during *Disparity Estimation*. The *Search Range Calculation* limits this disparity search range to the maximal required range, making the method faster. Another advantage of limiting the search range, albeit a minor one, is that disparity estimation becomes more accurate because some incorrect vectors which could lead to a local minimum in the error measurement are not investigated and so do not influence *Disparity Estimation* itself.

Dynamic programming makes it possible to select the optimal disparity vectors from all possible disparity vectors within the search range during *Disparity Estimation*. Two of the earliest publications on stereo using dynamic programming are [BB81] and [OK85]. More recently [GLY95] improved a disparity estimation system by taking into account occlusions. Further developments which involve performing disparity estimation based on arbitrarily shaped 3-D regions and also considering an extended neighbourhood are described in [Fal94] and [FS95].

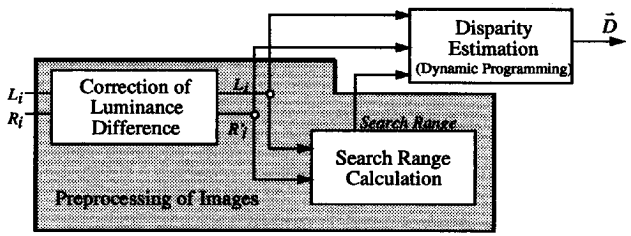


Figure 3.1: Disparity estimation system

Before going into detail about the application of dynamic programming to disparity estimation, a simple example will be discussed to explain the basic principles. In the following Sections, improvements to existing stereo disparity estimation algorithms that

assume knowledge of the epipolar geometry will be discussed. Further enhancements that could lead to a universal disparity estimator which does not require epipolar geometry information or preprocessing to improve disparity estimation, will be described. Finally the developed disparity estimation system will be evaluated experimentally.

3.2 Principle of Dynamic Programming

Dynamic programming [Bel57] is an optimisation process that chooses the globally optimal one from a number of possible solutions according to a pre-defined criterion, such as the lowest solution cost. The simplest and most straightforward application of dynamic programming is the determination of the *shortest path or route through a network* [Dan75].

Consider the (stylised) road map shown in Figure 3.2. A driver wishes to find the shortest route from point P to point Q. There are six intermediate junctions A, B,..., F. The lengths of all existing road sections connecting two points in the area are indicated on the map. Any unbroken chain of road sections starting at P and ending at Q represents a possible route through this network of roads.

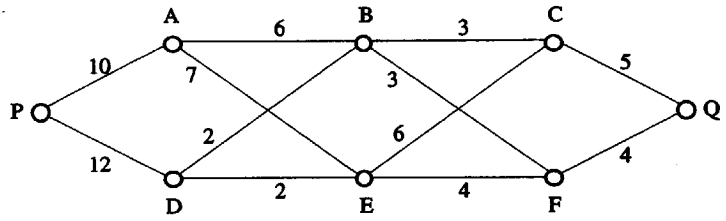


Figure 3.2: An example "road map" [Dan75]

Assume that the direction of travel is always from left to right. When the driver arrives at, say, point B, he never travels back to A or D, but proceeds either to C or F. The number of possible routes is, therefore, finite. The problem can then be solved by enumerating the alternative routes and comparing their total length.

Any route from P to Q is the result of three successive decisions. Starting at P, the driver must decide whether to go to A or to D. If he chooses to drive to A, he can then proceed either to B or E, and so on. The number of possible paths P_{DP} can be calculated as

$$P_{DP} = (number\ of\ alternatives)^{(number\ of\ consecutive\ decisions)} \qquad (3.1)$$

Since each decision is a choice between two alternatives and there are three consecutive decisions to be made, there are $2^3 = 8$ possible combinations, i.e. 8 possible routes. This can be illustrated graphically using the decision tree in Figure 3.3. The root represents the starting point P and the branches the road sections (length indicated). Comparing the total lengths from root to leaves, it can be seen that PDBFQ is the shortest route.

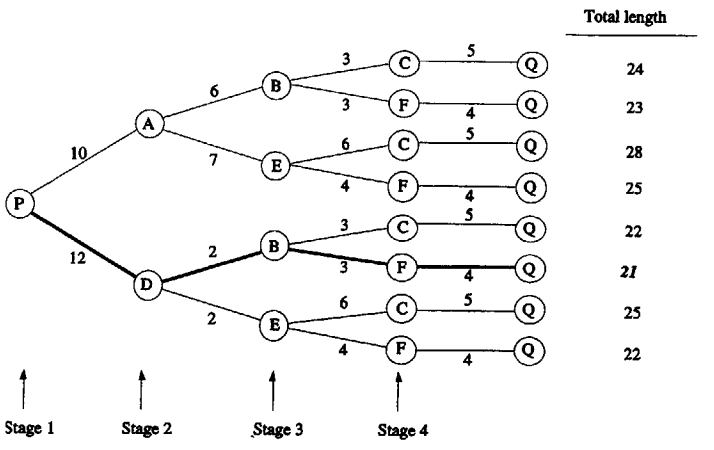


Figure 3.3: Graphical illustration of the decision tree [Dan75]

This optimal solution is found by applying *Bellman's principle of optimality* [Bel57]: If the optimal route from P to Q passes through B, the remaining part of the route (from B to Q) must also be optimal. The optimal route from P to Q cannot for instance contain BCQ because there is a shorter route from B to the destination, namely BFQ.

The problem can then be solved using backward recursion. Starting at point Q, the optimal solution has to contain F at stage 4, as the distance from F to Q is the minimal one. In the next step, the optimal solution at stage 3 is searched for. It now has to contain F and the solution is BFQ with the minimal distance 7.

As can be seen from this simple example the theory of dynamic programming is based on a single concept of great power and simplicity. *Bellman's principle of optimality* basically says that an optimum decision policy has the property that any part of an optimum trajectory from an intermediate state to the final state is itself the optimal trajectory from the intermediate state. This makes it possible to determine a total optimum decision policy and a corresponding minimum cost function by starting at the end of the process and working backward one stage at a time whilst only considering the decision at that stage. When taking

this decision, the short-term cost at the stage in question and the long-term consequences of having to follow the optimal policy from the next state to which this decisions leads have to be considered. In practice, this leads to a cost matrix which represents the costs for a certain state and a particular decision. After making this sweep backward through the stages, the optimum decision sequence and the optimum trajectory can be determined by sweeping forward through the states and determining the next decision, depending on the accumulated values of the cost matrix. This cost matrix for the example in Figure 3.2 is depicted in Figure 3.4. In the backward search for the best solution at each stage the accumulated values will be calculated and stored in the elements of the cost matrix. These values are printed in bold in Figure 3.4. Sweeping forward through the states - following the optimal path - will result in the final solution. The full scope of dynamic programming is described in [LC78], for example.

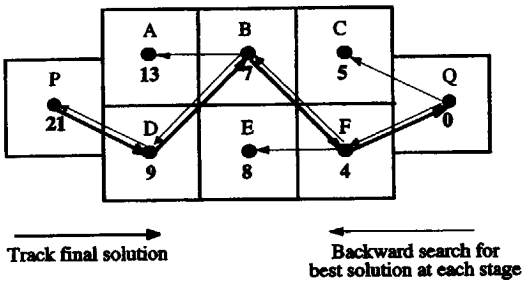


Figure 3.4: Cost matrix for the example in Figure 3.2

3.3 Constraints for Disparity Estimation

Disparity estimation is governed by several constraints. They fall into two groups: physical constraints - due to the camera geometry, such as *epipolar lines, orientation* and *uniqueness* - and model assumptions such as *continuity* and *ordering* which come from the optimisation method chosen in this thesis.

1. *Epipolar lines: Corresponding image points must lie on corresponding epipolar lines.* As described in previous Chapters, knowledge of the epipolar geometry is essential for most of the approaches adopted for disparity estimation. However, one of the goals of this Chapter is to eliminate this constraint and to develop a general purpose disparity estimator.

2. *Orientation: The disparity $\vec{D}$ within an object cannot exceed a certain value $\vec{D}_{max}$ or be lower than $\vec{D}_{min}$ which depends on the stereo system parameters.* According to constraint 1 (epipolar lines), a point A'_l must correspond to a point on the corresponding epipolar line in the right image (see Figure 3.5). It is possible however, due to the orientation of the object in space, that the disparity $\vec{D}$ varies considerably. This disparity range is analogous to the size of Panum's area [Pan1858] for the human visual system which is the disparity region in the retina where fusion occurs. The possible corresponding point for point A'_l on the epipolar line therefore can only lie on the line segment from A'_{lr} to A'_{2r}. As, according to this constraint,- there is a maximum disparity, the search area in disparity estimation can be limited to this maximum.

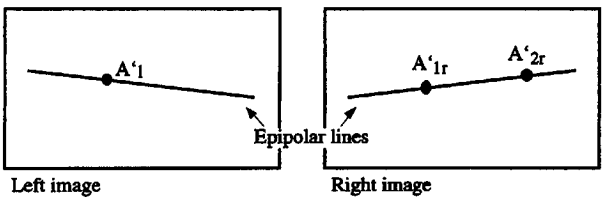


Figure 3.5: Possible disparity range within an object in a stereo pair

3. *Uniqueness: For each image point in one image of a stereo pair there is at most one corresponding point in the other.* Normally there is a one-to-one projection in both stereo images. However, this does not apply if there is occlusion.

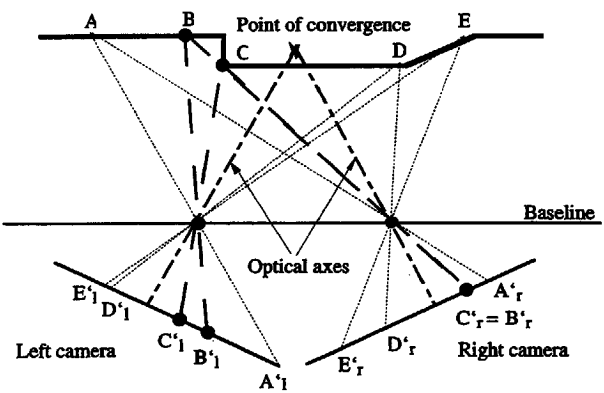


Figure 3.6: Occlusion of BC in the right image

In Figure 3.6, points B and C, correspond to points B'_l and C'_l in the left image, but in the right image they are seen superimposed as one point due to occlusion. Therefore, projections of points between B and C exist in the left image, but there is no corresponding point in the right image. Uniqueness will be the key to detecting occlusions in the estimator.

4. *Continuity: Inside an object, disparity is continuous and disparity discontinuities are only allowed at object boundaries.* Since surface changes are usually small compared to the viewer's distance, except at depth discontinuities, piecewise smooth depth can be imposed. As there is simple relation between disparity and depth (equation (2.7)), a piecewise smooth disparity can be assumed.

5. *Horizontal ordering: Image points on corresponding epipolar lines should have the same horizontal order.* Figure 3.7 shows an example where the order of points A, B, C is the same in both the left and right images. This constraint is needed when dynamic programming is applied as the matching pattern (left epipolar line) has to be ordered in the same way as the reference pattern (right epipolar line).

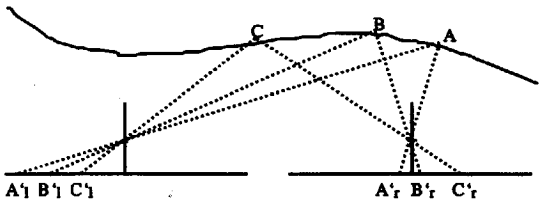


Figure 3.7: Order of consecutive points is the same in both images of a pair

However, there is a special case, depicted in Figure 3.8, where the ordering constraint is not satisfied; this is referred to as the "double-block illusion" [GLY95]. In such situations, it seems the human visual system attempts to fit the data to two surfaces obeying the ordering constraint and hence obtains transparency [GB88]. Two other theoretical solutions are either to mismatch the two objects by using the ordering constraint (due to the reversal of order in the two images e.g. point D'_l in the left image will be matched to point B'_r in the right image), so causing the sensation of two tilted planes, or to match just one object (considering the other one occluded), so causing the sensation of two occluded regions - one to the left and the other to the right.

The constraints described above will have a large impact on disparity estimation. Probably the most important constraint is *horizontal ordering*. As the driver was not allowed to go back in the example in Section 3.2, the points on corresponding epipolar lines have to be in the same order.

Otherwise, as shown in Figure 3.8. Bellman's principle of optimality will not be applicable, disparity estimation will then fail and either detect an occluded area or give wrong results.

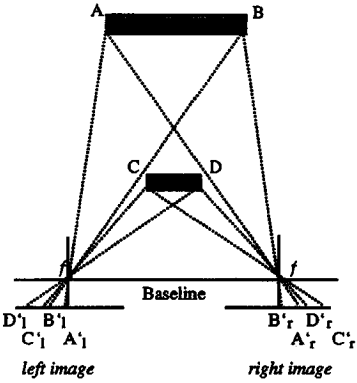


Figure 3.8. Double-block illusion [GLY95]: the ordering constraint is not satisfied

Second, *uniqueness* is a system requirement. Assuming that corresponding points are ordered and unique, the described approach will be able to detect occluded areas as indicated in Figure 3.6. *Continuity* is the key to detecting depth discontinuities, so allowing disparity discontinuities. Assuming piecewise smooth disparities, disparity discontinuities can only occur at object boundaries, where depth discontinuities occur. The *orientation* constraint, is a way of reducing the matching space and so decreasing the amount of computation; the same applies to the *epipolar line* constraint.

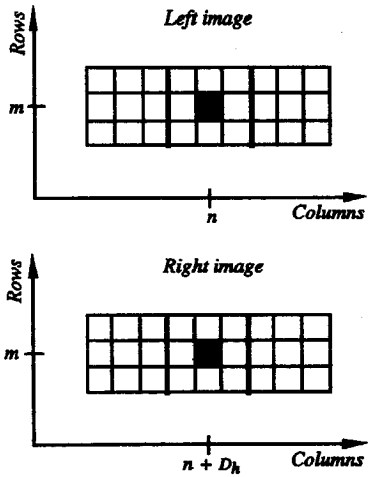
3.4 Dynamic Programming for Disparity Estimation with Known Epipolar Geometry

The general problem of finding correspondences between images involves searching within large parts of the image. The knowledge of the epipolar line geometry simplifies this image-to-image correspondence to a set of line-to-line correspondence problems. That is, once a pair of epipolar lines is calculated, the search for a pair of corresponding points in the left and right image can be confined solely to a well-defined searchline. This can be treated as the problem of finding a matching path on a two-dimensional search plane whose vertical and horizontal axes are the left and right epipolar line. A dynamic programming technique can handle this efficiently [GLY95]. In the following Section, the a priori matching costs and the matching space and then also an enhanced version which takes occlusions into account will be discussed.

3.4.1. A Priori Matching Costs

As in the example in Section 3.2, where the distance was given a priori as the cost of getting from one point to another, in general, dynamic programming needs these *a priori costs.*

As far as disparity estimation as described in this thesis is concerned, one goal is to find the best matches of a pixel P'_l in the left image to another P'_r in the right image. A *Normalised Feature Difference* (NFD) is used as a measure of the mutual correspondence between two pixels. A pixel-based NFD should be calculated for a pixel-based disparity map. Matching a single pixel can give misleading results, as matching will then result in a lot of possibilities with an identical NFD. Therefore, a matching window is selected around the pixel to provide a more matchable pattern for unique matching. In Figure 3.9, the matching window designed for the NFD calculation at a single pixel (n,m) is shown. The quality of the matches is very much dependent on the window size. A small window provides a small matching area which can be fitted to many patterns in the other image and it can produce a spurious disparity vector. A large window would be desirable, but a major limitation is the possibility of getting incorrect estimates near depth discontinuities. The window that is used is rectangular so as to allow a good match along the epipolar line and to allow pixels from above and below the epipolar line to contribute to the NFD.



D_h: horizontal component of disparity vector

Figure 3.9: Window used to calculate the NFD for image point (n,m) [GLY95]

First of all, block-matching evaluates the *Feature Difference* (FD) for each possible match with every possible disparity vector $\vec{D}$ in the search area. This feature difference is subsequently normalised (NFD) to give a value between 0 and 1 representing the a priori costs and to be integrated in a total cost function evaluated during optimisation (see Section 3.4.4). According to Figure 3.9, point (n,m) in the left image is displaced by the disparity D_h in the right image. The feature difference $FD_{n,m}$ for point (n,m) with disparity vector $\vec{D}_{n,m}$ is given by the formula

$$FD_{n,m}(\vec{D}_{n,m}) = \sqrt{\sum_{i=n-\tau}^{n+\tau}\sum_{j=m-\upsilon}^{m+\upsilon}\left(W_L(i,j) - W_R(i+D_h,j)\right)^2} \qquad (3.2)$$

where:

n,m	pixel coordinates
D_h	horizontal component of the disparity vector
$2\tau + 1$	width of the window in (horizontal) x-direction
$2\upsilon + 1$	width of the window in (vertical) y-direction
$W_L(i,j)$	luminance at (i,j) in the left image
$W_R(i,j)$	luminance at (i,j) in the right image

This feature difference is derived from the squared luminance difference, as can be seen in equation (3.2). The $NFD_{n,m}$ for the point *(n,m)* with disparity vector $\vec{D}_{n,m}$ is calculated from the formula:

$$NFD_{n,m}(\vec{D}_{n,m}) = \frac{c}{N} FD_{n,m}(\vec{D}_{n,m}) = \frac{c}{N}\sqrt{\sum_{i=n-\tau}^{n+\tau}\sum_{j=m-\upsilon}^{m+\upsilon}\left(W_L(i,j) - W_R(i+D_h,j)\right)^2} \qquad (3.3)$$

where: c is a normalisation constant to yield values smaller than 1
$N = (2\tau + 1)(2\upsilon + 1)$ is the number of pixels in the search window

If the normalised feature difference is zero this means that the luminance of the two windows is the same at all points, whereas a high value of the normalised feature difference shows uncorrelated luminance.

With luminance values in the range of [0, 255], the theoretical maximum value that can be reached by FD is $\sqrt{N \cdot 255^2}$. In order to guarantee NFD ≤ 1, c therefore should be $\frac{N}{\sqrt{N \cdot 255^2}}$.

However, in practice this value never will occur. Empirical studies have determined that the following parameter values are suitable for disparity estimation:

$$c = \frac{1}{12},\ \tau = 4,\ \upsilon = 1 \tag{3.4}$$

Actually these parameters do not heavily influence the results. Other values can still be accepted, nevertheless disparity estimation delivers suitable results. The values of the calculated NFD now will be used as the costs in the dynamic programming approach.

3.4.2 The Matching Space

Given a point in the left image of a stereo pair, the corresponding point in the right image lies on the epipolar line. The search for this corresponding point can, therefore, be restricted along this epipolar line. Figure 3.10 illustrates this principle. For each point with index n on the epipolar line in the left image, the corresponding point with index m on the right epipolar line is searched for. The disparity D_h for a point (n,m) then can be written as $D_h = m - n$, which is the vertical distance from the diagonal in Figure 3.10. Points on the diagonal itself, therefore, indicate the zero disparity. The two-dimensional space spanned by the two corresponding epipolar lines is called the matching space. The disparity estimation process can be thought of as looking for a path through this matching space from the bottom left corner to the top right corner for which the accumulated cost value ACV_m shown in equation (3.5) is the smallest.

$$ACV_m = \sum_{n=1}^{n_{max}} NFD_{n,m} \tag{3.5}$$

Dynamic programming - as an optimisation method - will find the optimal path among all possible paths.

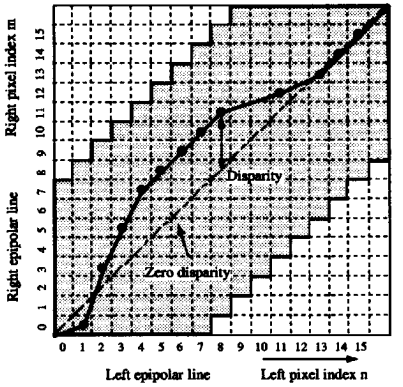


Figure 3.10: An illustrative path through the two-dimensional matching space

Based on the a priori costs of the disparity vectors per pixel, the dynamic programming searches for the optimal solution for each epipolar line. The a priori costs are stored in the elements of the matching space. In Figure 3.10, the matching space was also reduced (as indicated by the grey area) by limiting the possible disparity values, based on the orientation constraint. In the depicted case, the maximum possible disparity value was set to ±7 pixel, the minimum cost path now has to be found within the grey area. Figure 3.11 shows a part of the matching space depicted in Figure 3.10 in node form. Again the axes are the epipolar lines, the nodes now show the disparity values.

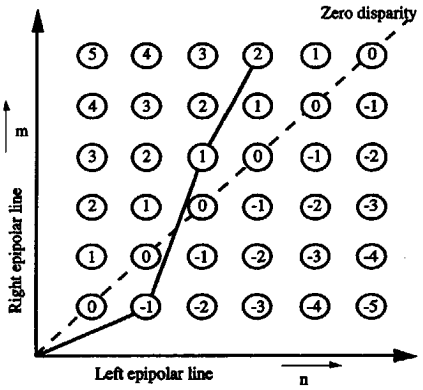


Figure 3.11: Matching space as a set of nodes

As described in Section 3.2, the search for the optimal path through the matching space starts with the pixel at the right-hand end of the epipolar line. Due to the "horizontal ordering" constraint, the number of possible predecessors in the backward search for the best solution at each stage is limited: For a match *(n, m)* possible predecessors can be found either in row *(n+1)* - i.e. looking for the best match for the next pixel - or on line *m* - looking for occluded pixels which can be seen in the left image but not in the right one - as shown in Figure 3.12. In this figure, the numbers of the elements indicate the disparity difference of the current element (dark grey) to the possible predecessors. In the second pass of the dynamic programming procedure, when searching for the final solution, these occlusions are indicated by a "negative" disparity jump, i.e. a jump from a larger disparity value to a smaller one. In such a case, a number of pixels on the epipolar line equivalent to the size of the disparity jump will be omitted, their costs will not be added to the total costs of the path. As will be seen later, this poses significant problems which will be taken care of in Section 3.4.3.

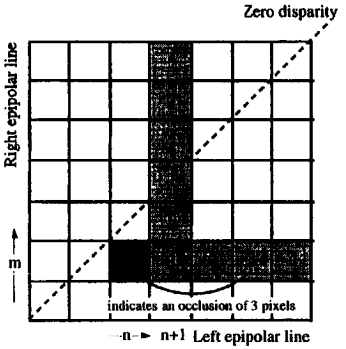


Figure 3.12: Possible predecessors when looking for the best partial solution

In Figure 3.13, the matching space and the ideal solution (the drawn path) for the example in Figure 3.6 are shown. The matching space depicts the correspondence of points A'_l to E'_l and A'_r to E'_r. Point A'_p for example, has a corresponding match at point A'_r. There is no correspondence for points between B'_r and C'_r on the right epipolar line and so consequently the path through the matching space will be interrupted. This interruption indicates a disparity discontinuity and, therefore, exhibits an occlusion. The diagonal line again is the *zero disparity line* as can be seen in the previous figures too.

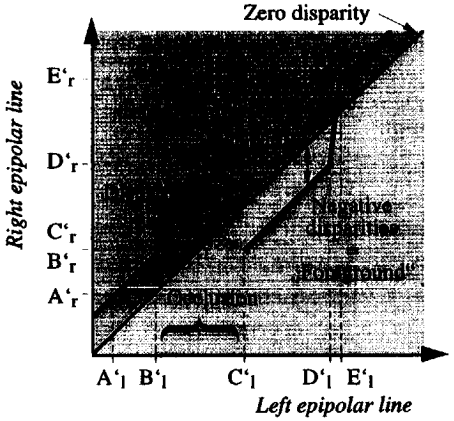


Figure 3.13: Matching space

The disparity value between a point on the left epipolar line and its match on the right epipolar line is the vertical distance between the zero disparity line and the plot of the path through the matching space, as explained above. This means Figure 3.13 is easy to interpret:

Starting at the left edge of the image, positive disparity values have been found, which means that the points from A to B lie behind the point of convergence of the cameras (see Figure 3.6). Next, on one axis B'_r and C'_r are coincident, whereas B'_l and C'_l are not on the other: an occlusion in the right image is the reason. Points from C to D are then in front of the point of convergence. The connection from D'_l to E'_l is smaller than that from D'_r to E'_r, a foreshortening as introduced in Chapter 2 has occurred. This happens because the plane is not parallel to the baseline. Also in this example the crossing of the zero disparity line indicates, that points are now behind the point of convergence.

The following example explains the process of dynamic programming with real image data. These data are taken from the synthetically produced "Mirror" sequence [Fra96]. Table 3.1 shows the grey values for three lines of the left image, Table 3.2 the corresponding grey values in the right image. The pixels of interest are surrounded by a bold rectangle and marked with numbers from 1 to 6. The grey values in the table were taken from an area with disparity zero. As noise with a level of 30 dB was added to this sequence, the values are not identical in the two tables.

113	111	114	111	110	108	111	101	114	117	113	121	125	139
114	111	109	112	104	117	106	99	107	111	115	110	106	114
113	111	120	97	110	117	104	113	108	114	111	101	104	111
				1	2	3	4	5	6				

Table 3.1: Real grey values from left image

107	105	122	120	117	104	108	104	105	112	112	116	125	125
107	108	115	105	113	101	108	98	102	121	110	118	114	111
114	113	115	113	95	113	112	109	102	109	114	107	101	105
				1	2	3	4	5	6				

Table 3.2: Real grey values from right image

The matching space for this example is shown in Figure 3.14. The entries for the elements of Figure 3.14 (a) are the NFD calculated according to equation (3.3) with the parameters of (3.4). For the sake of simplicity, only a small part of the whole matching space is depicted, namely for pixels 1 to 6 with bold highlighting. Also, the search area has been restricted to ±3 pixels in this example and element (6, 6) has been forced to be part of the best solution. Figure 3.14 (b) shows the accumulated cost values along the chosen path printed in bold.

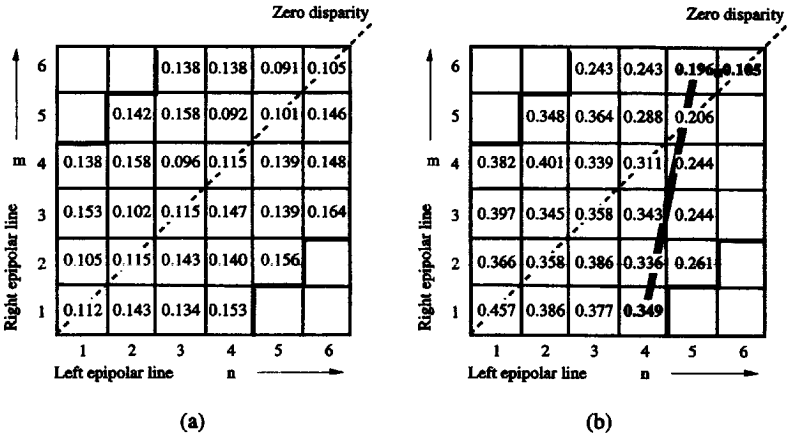


Figure 3.14: Matching space filled with NFD from real data (a) and with accumulated cost values (b)

The bold path is found to be the best possible according to the rule in Figure 3.12. As the values of the NFD are rather similar, a wrong path was selected as the optimal one, leading to disparity discontinuities where there should be continuous zero disparity.

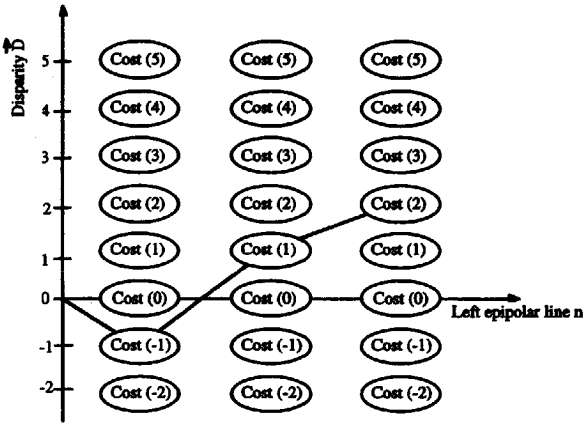
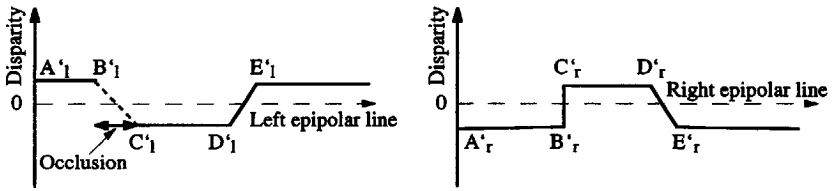


Figure 3.15: Matching space, spanned by the left epipolar line and the disparity

Because dynamic programming only runs on single image lines, an equivalent representation of the graph shown in Figure 3.11 - where the axes are the left epipolar line and the disparity values - is also possible. When this approach is adopted, the costs are the value of a node, described by the disparity and the point in the left image. Figure 3.15 shows a matching space representation of this kind which can be transformed to the "standard" cost matrix.



(a) (b)

Figure 3.16: Matching spaces, spanned by the left epipolar line and the disparity (a) and the right epipolar line and the disparity (b) for the example of Figure 3.6

Figure 3.16 shows the same representation as Figure 3.15, spanning the matching space with the left epipolar line (Figure 3.16 (a)) and the right epipolar line (Figure 3.16 (b)), respectively. In Figure 3.16 (a), the occlusion in the left-to-right disparity estimation between points B'_l and C'_l is again seen as the dashed line. As with Figure 3.13 this occlusion can be detected as a "negative" disparity jump. In Figure 3.16 (b), representing the right-to-left estimation, a positive disparity jump - without occlusion - between points B'_r and C'_r can be seen.

The algorithm was tested on a synthetic test sequence called "Mirror" [Fra96]. This sequence is a composite of the well-known scenes "Clown " and "Lena". In this composite, the main part of the scene is at zero disparity, while the view of Lena through the mirror has a disparity equal to the image number (images being numbered from zero). In the odd numbered images "Lena" has been moved one pixel to the left compared to the previous odd numbered image. In the even numbered images "Lena" has been moved one pixel to the right. In this way the motion is introduced in every second image, the face of "Lena" therefore stays visible in the whole sequence. Occlusions occur at the edges of the mirror, for example in Figure 3.17 the feather on Lena`s hat is visible in the left view, but not in the right view. 30dB Gaussian noise was added to the sequence used in this thesis to make it easier to assess how the algorithm would handle real images.

Figure 3.17: Synthetic test sequence "Mirror", left and right image No. 32

Figure 3.18 shows the result of a left-to-right view disparity estimation performed using the NFD as the only criterion. Zero disparity is shown as value 128, so being depicted as grey. Occlusions are shown in black. As can be seen, this result is not at all satisfactory. The problem is that whenever a disparity jump of size n is detected, the next n pixels in the line will be treated as occlusions. Therefore, the costs of these pixels will not influence the costs of the path which make a jump attractive. Also, adding noise made the Normalised Feature Difference, NFD, optimal for spurious vectors most of the time. This means that only a few disparity values are estimated and most of the image is treated as an occlusion.

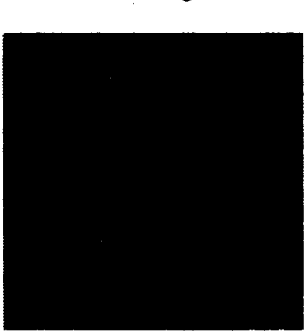
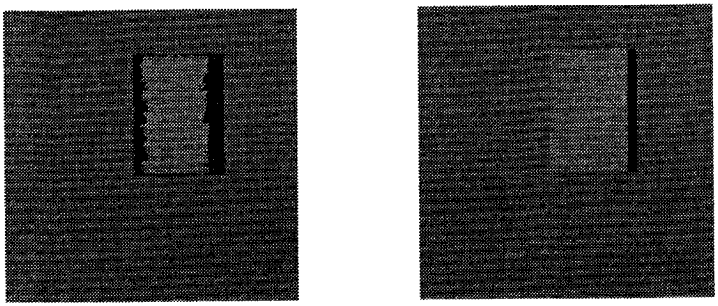


Figure 3.18: Result of disparity estimation of "Mirror" (image 15) using a priori costs only

In another version of this test sequence without noise, this problem did not occur as the NFD was a minimum for the correct vectors. This can be seen in Figure 3.19 (a), where the background of the image is correctly matched with disparity 0 and the image of Lena with disparity +15.



(a) (b)

Figure 3.19: Result of left-to-right disparity estimation of "Mirror" without noise (image 15) using a priori costs only (a) and artificially generated correct vector field (b)

Even in this noiseless case of Figure 3.19 (a) the result is not satisfactory compared to the correct vector field depicted in Figure 3.19 (b). Again problems occur at edges with disparity discontinuities. This can be avoided by taking these discontinuities into account in the costs used for dynamic programming. If discontinuities are only allowed where useful (preferably at object boundaries), a kind of smoothing in areas without edges will be achieved.

3.4.3. Smoothing the Vector Field

Up to now only the Normalised Feature Difference (NFD) was used as the cost for the dynamic programming. With noisy images, disparity jumps cause a lot of problems due to local minima. However, where occlusions occur, disparity jumps have to be allowed. Therefore, straightforward smoothing of the vector field is not possible. To take occlusions into account a second cost will be added in order to allow these disparity discontinuities at the appropriate places.

According to the fourth constraint "Continuity" defined in Section 3.3, piecewise smooth disparity inside an object can be assumed. Disparity jumps will occur either at an object boundary where there is a depth difference (e.g. between the object and the background) or due to noise in the vector field. In the latter case, assuming piecewise smooth disparity, the disparity values can be smoothed, whereas in the first case the disparity jump has to be maintained. For that reason, a cost function is introduced, which penalises disparity jumps according to their value. The higher the value of the disparity jump, the higher is the probability that a real jump at the boundary of an object exists, whereas with small jumps the reason is probably noise. The additional costs will be added to the costs already derived by the a priori statistics described in the previous Section. The combination of both gives the final cost function described in the next Section.

If a possible disparity vector $\vec{D}$ at point *(n,m)* is denoted by $\vec{D}_{n,m}$ then the disparity jump δ between disparities of consecutive points n and $n-1$ along the epipolar line m is defined as:

$$\delta = \left|\vec{D}_{n-1,m} - \vec{D}_{n,m}\right| \tag{3.6}$$

It is necessary to use the disparity difference between adjacent pixels so as to be able to decide whether a "real" disparity jump exists or simply noise. The function $f(\delta)$ is required to make this decision. Therefore $f(\delta)$ should have the following characteristics:

- $f(\delta) > 0$. The minimal cost of a disparity vector is the normalised feature difference NFD. As $f(\delta)$ increases to NFD, it is not allowed to have a negative value, which then would reduce the original costs described by the normalised feature difference.

- The behaviour of $f(\delta)$ should be different for small and large disparity jumps. The objective of $f(\delta)$ is to reduce noise in the vector field but also to allow jumps at disparity discontinuities. A large jump, therefore, should only be possible if the NFD of the disparity vector is a lot better than for a small jump or even no jump at all. If there are, say, two different disparity vectors with the same NFD, one forcing a jump and the other having the same value as the adjacent one, it would better to use the latter one, so smoothing the vector field.

  $f(\delta)$, therefore, should sharply increase the costs for small jumps, bearing in mind the smoothing of small jumps. On the other hand, the increase should be rather stable for large jumps, allowing jumps only where feasible. As each disparity jump results in an occlusion, large jumps should only be allowed where it is reasonably certain that the detected jump is correct. Therefore, the costs for large jumps will be set to a high value in comparison with the NFD.

Now, the problem is to find a function that best describes this behaviour. Various authors have proposed different functions. In [YGB90], for example, a logarithmic function (depicted in Figure 3.20) is proposed:

$$f(\delta) = ln(1+e) - ln\left(1+e^{1-\delta^2}\right) \tag{3.7}$$

As argued above, it is important to determine how the cost-function behaves for small and large disparity jumps δ. The function proposed in [YGB90] rapidly increases to a value greater than that of the defined NFD. Even jumps greater than 1 will be penalised very heavily, which gives a very smooth disparity vector fields.

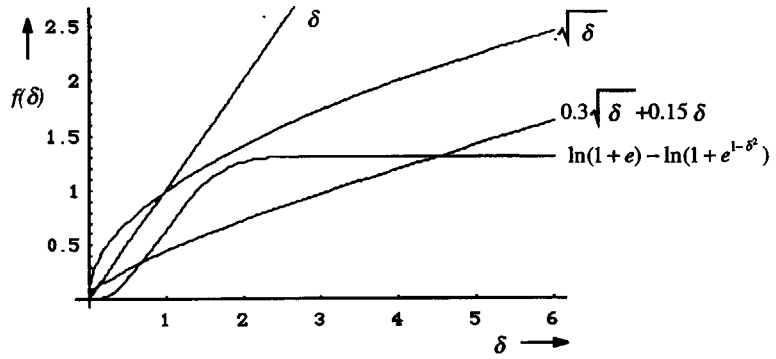


Figure 3.20: Various suggestions for the cost-function f(δ)

On the other hand, large jumps will be treated almost in the same way (see Figure 3.21), so that it is not possible to distinguish between large jumps of different sizes. This leads to a behaviour which, in the final analysis, means that the NFD would again be the only decisive criterion. To avoid this, the chosen function should increase the penalty in relation to the size of the jump.

A function that gives a sufficiently high gradient for small x and an approximately constant penalty increase, is the root function (Figure 3.20). The function proposed for disparity jumps in [GLY95] is:

$$f(\delta) = \mu \cdot \sqrt{\delta} + \varepsilon \cdot \delta \tag{3.8}$$

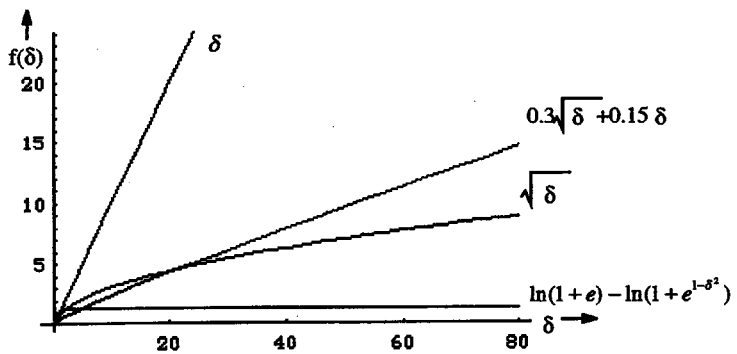


Figure 3.21: Cost functions for large disparity jumps

In [GLY95] it is stated that function (3.8) gives reasonable results with μ=0.3 and ε=0.15. This function gives acceptable costs for both small and large jumps. Smaller values of μ and ε would give a penalty which is too small compared with the NFD; larger values would give rise to a situation where the NFD could not influence the result anymore. Therefore, the penalty function used for disparity jumps in this thesis is:

$$f(\delta) = 0.3 \cdot \sqrt{\delta} + 0.15 \cdot \delta \tag{3.9}$$

Unlike the NFD, the disparity jump penalty does not take luminance into account, it relies only on the size of the possible disparity jump. It will, therefore, have a great influence on the chosen path, as the size of the disparity jumps depend on the path through the matching space.

3.4.4. Combining the Cost Functions

The normalised feature difference NFD (Section 3.4.1) is a criterion that the block-matching procedure uses to assess the quality of the matching. The disparity jump cost (Section 3.4.3) gives the size of the penalty for any disparity jump. A combination of the two functions can be used to obtain decisions which are better than decisions based solely on one function. By adding up equations (3.3) and (3.9), a cost function CF is obtained that forces smoothness, handles disparity jumps and takes the NFD into account.

$$CF_{n,m}(\bar{D}_{n,m}) = \frac{c}{N}\sqrt{\sum_{i=n-\tau}^{n+\tau}\sum_{j=m-\upsilon}^{m+\upsilon}\left(W_L(i,j) - W_R(i+D_h,j)\right)^2} + 0.3 \cdot \sqrt{\delta} + 0.15 \cdot \delta \tag{3.10}$$

Dynamic programming is now used to evaluate all possible paths through the matching space by adding up the matching costs defined in equation (3.10). The best path, in other words the one with the smallest accumulative costs, is then chosen.

The value ACV is the accumulated cost value calculated for a specific path along the epipolar line m and is defined by equation (3.11):

$$ACV_m = \sum_{n=1}^{n_{max}} CF_{n,m} = \sum_{n=1}^{n_{max}}\left(NFD_{n,m} + f(\delta)\right) \tag{3.11}$$

This principle will be illustrated using the real data from Tables 3.1 and 3.2. Table 3.3 shows the jump costs $f(\delta)$ (equ. (3.9)) as a function of the jump size δ.

δ	0	1	2	3	4	5
$f(\delta)$	0	0.450	0.724	0.969	1.20	1.421

Table 3.3: Jump costs $f(\delta)$ given by equation (3.9)

These values are added to the NFD-values (Figure 3.14 (a)) of the candidates of the previously chosen value (bold). This is depicted in Figure 3.22 (a). Figure 3.22 (b) again shows the accumulated cost value, this time taking the jump costs into consideration.

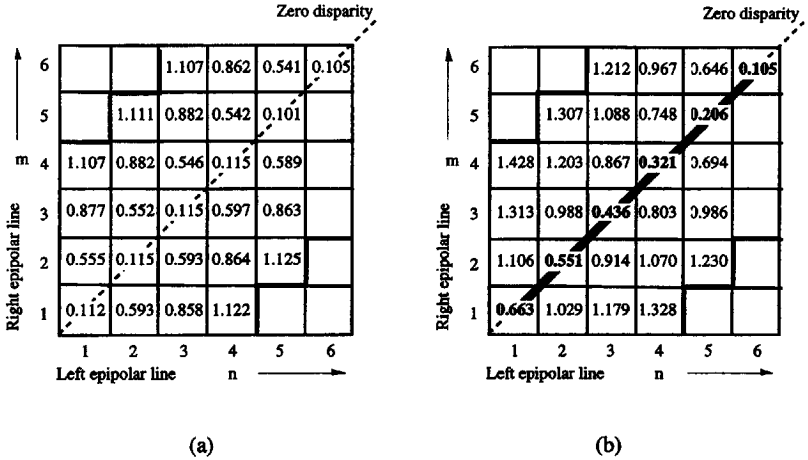


Figure 3.22: Matching space filled with cost function $CF_{n,m}$ from real data from tables 3.1 and 3.2 (a) and accumulated cost value ACV_m (b)

By using this combined cost function a different optimal path was found. In this example, all jumps have been avoided so making it possible to find the correct path through the matching space. Figure 3.23 shows the result for the "Mirror" image. Compared to Figure 3.19 (a) the result looks better now. The disparity jumps have been detected more precisely, this way improving the overall quality of the estimation.

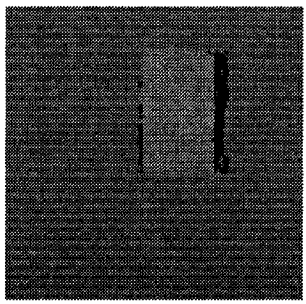


Figure 3.23: Disparity vector map for image-pair 15 of the test sequence "Mirror" using cost function $CF_{n,m}$

3.5 Dynamic Programming for Disparity Estimation without Known Epipolar Geometry

The principle described in the previous Section is the standard version of dynamic programming for a two-dimensional matching space. If it is to be employed for disparity estimation, the epipolar geometry must be known. As the disparity estimation method developed in this thesis is intended for the general use without any knowledge of the epipolar geometry, some modifications have to be made. In the following Section, these modifications will be discussed.

3.5.1 Enlargement of the Search Area

Without knowing the epipolar geometry, the correspondence problem can no longer be solved with a one-dimensional search. As with motion estimation, a two-dimensional search area has to be introduced. As already discussed in Section 2.1.2, the vertical deviation of the epipolar line from a horizontal line is less than or equal to two pixels in the test sequences used in this thesis. To obtain reliable disparity vectors, this deviation has to be included in the estimation.

Figure 3.24 shows the main difference between this case and the case where the epipolar geometry is known. In the original case, as described in the previous Section, the search is only performed along the epipolar line. In the new general case, the search is performed along a horizontal line, but the possible vertical deviations from this line are also taken into account.

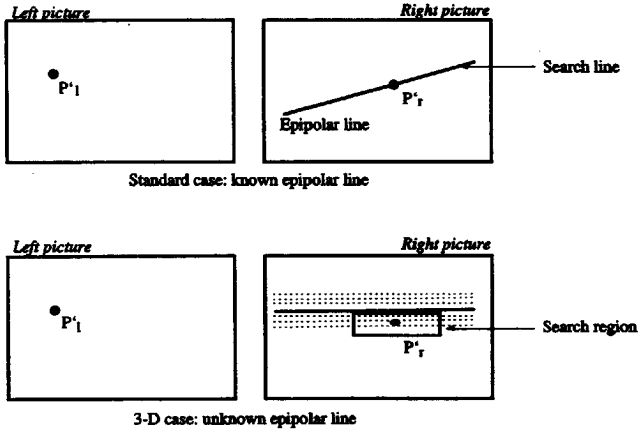


Figure 3.24: 3-D disparity estimation versus standard approach

This leads to a two-dimensional search region for each pixel. Disparity now is the horizontal component of the estimated displacement vector, while the vertical component is only considered as a deviation.

Ideally, this gives a two-dimensional disparity vector $\vec{D}$ whose horizontal component D_h is equal to the disparity vector that would be obtained if the epipolar lines were known, and whose vertical component D_v is the deviation of the vector with respect to the horizontal search line. If the epipolar lines were horizontal, the deviation would be zero, and the horizontal component D_h would be identical to the disparity vector as estimated using the method described in Section 3.4.

3.5.2 A Priori Matching Costs in a Two-Dimensional Search Area

The feature difference and normalised feature difference introduced in equations (3.2) and (3.3) have to be generalised to include the second dimension of the disparity vector. Equations (3.12) and (3.13) give a generally usable feature difference *GFD* and the resulting normalised feature difference *GNFD*, where the luminance difference now is calculated taking the vertical deviation d_v into account.

$$GFD_{n,m}\left(\vec{D}_{n,m}\right) = \sqrt{\sum_{i=n-\tau}^{n+\tau}\sum_{j=m-\upsilon}^{m+\upsilon}\left(W_L(i,j) - W_R\left(i+D_h, j+D_v\right)\right)^2} \tag{3.12}$$

$$GNFD_{n,m}\left(\vec{D}_{n,m}\right) = \frac{c}{N} GFD_{n,m}\left(\vec{D}_{n,m}\right) \tag{3.13}$$

As the vertical deviations will be quite small, the window defined in Figure 3.9 can also be used to calculate the general usable normalised feature difference. These normalised feature differences will be used in the dynamic programming as a priori costs.

3.5.3 The Three-Dimensional Matching Space

In addition to the matching space defined in Section 3.4.2, a vertical search now has to be added in order to take into account the vertical deviations. This results in a three-dimensional matching space as shown in Figure 3.25. The path-finding algorithm is extended to a 3-dimensional one by adding a vertical deviation axis to the matching space along which an additional vertical search is accommodated. This extension makes knowledge of the epipolar geometry superfluous. The only problem that arises is calculation time because of the much larger matching space. According to equation (3.1), the number of possible paths P_{DP} in this disparity estimation environment is equal to

$$P_{DP} = (number\ of\ possible\ disparities)^{(number\ of\ pixels\ per\ line)} \tag{3.14}$$

When vertical displacement is added to the matching space, the number of possible paths will increase exponentially. A method of reducing the matching space will be discussed in Section 3.6.2.

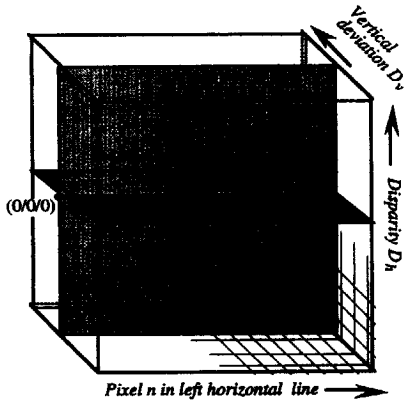


Figure 3.25: 3-D matching space

Figure 3.25 shows the extended three-dimensional matching space. Every node (n, D_h, D_v) represents a specific displacement vector (D_h, D_v) for the image point (n,m). The light shaded area of Figure 3.26 is where this vertical deviation D_v is zero. The dark shaded area is where the disparity D_h is zero. The same rules are used for the two-dimensional case and for the possible paths through this three-dimensional network.

The algorithm was tested with the "Aqua" test sequence [DIS92] as shown in Figure 3.26. Due to the interlaced format of the sequence, the algorithm was run on single fields instead of frames.

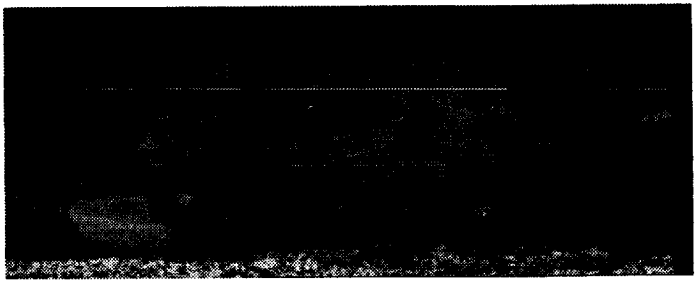


Figure 3.26: First field of the "Aqua" test sequence

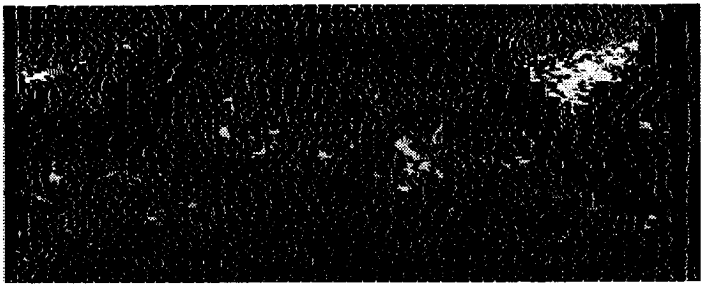


Figure 3.27: Disparity vector map of "Aqua" using a priori costs only

The problems encountered when the epipolar geometry is known (Figure 3.18) are also present and shown in Figure 3.27. A logical step would be to use vector field smoothing in this case too.

3.5.4 Smoothing in the Three-Dimensional Matching Space

As with two-dimensional matching space, disparity discontinuities will give useful information as to where occlusions occur. For these occlusions only a horizontal disparity jump is important. However, in order to find a more precise definition of the path through the matching space, vertical deviations will be taken into account too. Therefore, a new general usable jump function $Gf(\delta_h, \delta_v)$ is defined, which also takes care of the vertical deviations:

$$Gf(\delta_h, \delta_v) = f(\delta_h) + f(\delta_v) \tag{3.15}$$

δ_h being the disparity jump defined in Section 3.4.3 and δ_v the vertical deviation of point (n, m) with respect to its predecessor. As the deviations are usually very small, $f(\delta_v)$ will provide smoothing, allowing deviations only where really necessary. With a maximum deviation size of 4 pixels, f(δ) as defined in equation (3.9) is also suitable.

The total cost function *GCF* for a node (n, D_h, D_v) in the matching space is calculated in line with equation (3.10) as sum of equations (3.13) and (3.15):

$$GCF_{n,m}(\vec{D}_{n,m}) = GNFD_{n,m}(\vec{D}_{n,m}) + Gf(\delta_h, \delta_v) \tag{3.16}$$

with $$Gf(\delta_h, \delta_v) = 0.3\sqrt{\delta_h} + 0.15\delta_h + 0.3\sqrt{\delta_v} + 0.15\delta_v$$

For dynamic programming in the three-dimensional matching space, the accumulated cost value *GACV*

$$GACV_m = \sum_{n=1}^{n_{max}} \left(GNFD_{n,m}(\vec{D}_{n,m}) + Gf(\delta_h, \delta_v) \right) \tag{3.17}$$

is used as the criterion for finding the best possible path through this matching space.

In Figure 3.28, the result of disparity estimation of this kind and equalised for convenience is shown for the "Aqua" sequence, where bright parts indicate pixels further away than dark parts of the image. Occlusions are drawn in black. In addition to the disparity, the vertical deviations, as shown in Figure 3.29 are estimated. In this Figure the bright area in the top left corner of the image indicates a vertical deviation of +1, the grey colour 0 and the dark area in the lower left corner a vertical deviation of -1. The occluded areas again refer to the disparity shown in Figure 3.28. With the information depicted in Figure 3.29 actually a first idea about the epipolar line geometry is given.

The disparity vector map shown in Figure 3.28 will be used for segmentation. However, the occlusion gaps and the line structures can be expected to cause problems. Therefore, postprocessing of the disparity vector map is necessary.

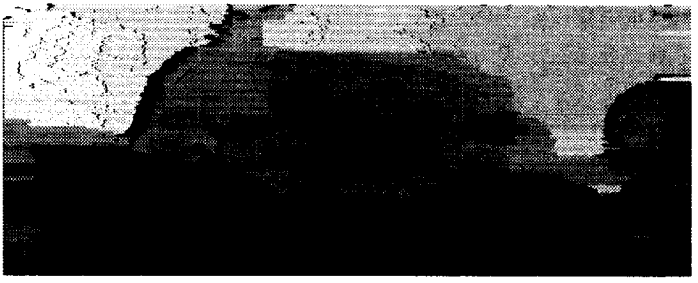


Figure 3.28: Disparity vector map for the "Aqua" test-sequence (equalised)

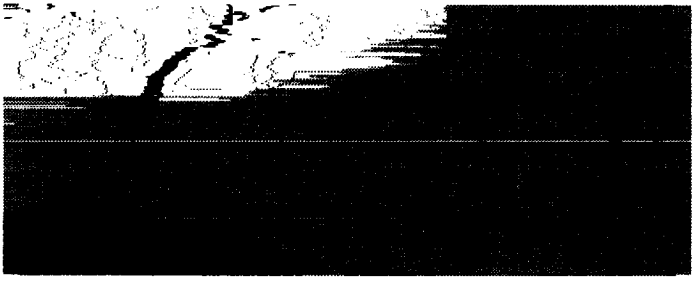


Figure 3.29: Vertical deviations in the "Aqua" test-sequence (expanded)

3.6 Preprocessing Images

As the stereoscopic sequences used in this thesis have been shot with two different cameras [DIS92, DIS94], the corresponding image pairs may have a luminance discrepancy [Fra92]. This is one of the most serious problems in disparity estimation and can lead to estimation errors [ZTT91]. The luminance balance compensation described in this Section improves matching.

A second topic is the large increase in computation made necessary by the generalisation to a three-dimensional matching space. In this Section, a method which limits this matching space is described. The goal here is to have the matching space "as small as possible, but as large as necessary". This will not only reduce the calculation cost, but also improve the quality of the matching.

3.6.1 Luminance Balance Compensation

In the test sequences [DIS92, DIS94], an imbalance between the luminance values of the left and right images is noticeable. Statistical modelling of these "imperfections" and appropriate compensation can improve the disparity estimation [ZTT91].

The model used to describe the difference in gain and offset settings is:

$$W_R(i,j) = a \cdot W_L(i - D_h, j - D_v) + b \tag{3.18}$$

where:		
	$W_L(i,j)$	luminance value at (i,j) in the left image
	$W_R(i,j)$	luminance value at (i,j) in the right image
	$(D_h, D_v)^T$	disparity vector
	a	amplifier gain
	b	offset

In this model it is assumed that a luminance value in the right image is derived from its counterpart in the left image (displaced by $(D_h,D_v)^T$ and therefore is at position $(i–D_h, j–D_v)$ in the left image) using an amplifier gain a and offset b.

Based on the model described in (3.18), a solution to compensate these imperfections is described in [Fra92]. Using the mean values ($\overline{W_L}$ and $\overline{W_R}$) and the variances (σ_L^2 and σ_R^2) of the luminance values of the two images - calculated as shown in equation (3.19) and (3.20) for the left image -

$$\overline{W_L} = \frac{1}{\pi \cdot \lambda} \sum_{i=1}^{\pi} \sum_{j=1}^{\lambda} W_L(i,j) \tag{3.19}$$

$$\sigma_L^2 = \frac{1}{\pi \cdot \lambda} \sum_{i=1}^{\pi} \sum_{j=1}^{\lambda} \left(\overline{W_L} - W_L(i,j) \right)^2 \qquad (3.20)$$

where λ the number of lines in the image

π the number of pixels per line

each pixel in the right image can be adjusted as in equation (3.21):

$$W_R'(i,j) = \hat{a} \cdot W_R(i,j) + \hat{b} \qquad (3.21)$$

where $W_R'(i,j)$ corrected luminance value of the right image point (i,j)

$\hat{a}$ $\sqrt{\frac{\sigma_L^2}{\sigma_R^2}}$

$\hat{b}$ $\overline{W_L} - \hat{a} \cdot \overline{W_R}$

This model ignores disparity and occlusions between the left and the right image. As long as these differences are not very serious this is not a problem. With the stereoscopic test-sequences [DIS92, DIS94] it can be assumed that applying this model will give satisfactory results, as the sequences have been shot in order to be viewed in stereo. The sequences therefore will be rather similar. By applying equation (3.21) to all right view pixels, the mean value and variance of the left and right image become equal. Some of the test-sequences exhibited luminance differences of such a magnitude that disparity estimation failed totally. Results from the DISTIMA project [Zie92a] showed that this compensation re-established a satisfactory function of the disparity estimators [Fra96].

3.6.2 Limiting the Matching Space

The three axes of the three-dimensional dynamic programming matching space are the pixels n of the horizontal line, the disparity axis D_h and the deviation axis D_v as depicted in Figure 3.25. The number of paths that have to be considered for calculation, increases exponentially with disparity and y-deviation. For the "Aqua" stereo sequence for example, the disparity search range was fixed at [-15, +30], the deviation search range was fixed at [-2, +2]. With 720 pixels per line, a matching space of 720x46x5 nodes has to be created in this example. Looking for the optimal path through this large matching space is computationally very expensive. Therefore, it is advisable to reduce this matching space.

A way of reducing the matching space is to pre-calculate the disparity search ranges for each line and consequently restrict the net of possible paths to within a useful range. The block *search range calculation* in Figure 3.1 performs this function.

It calculates the maximum and minimum disparity and deviation ranges in the pair. The algorithm consists of three steps (Figure 3.30). At first a large-block-matching disparity estimator using overlapping blocks with block-size 16x16 pixels gives an estimate of the range of the displacement vectors. Block-based median filtering follows to eliminate single errors in the vector field. The minimum and maximum search ranges are then calculated for each line to be used in the dynamic programming (only a single line in the image is scanned).

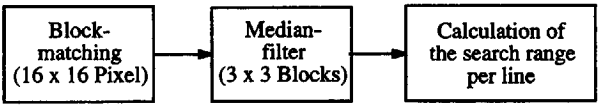


Figure 3.30: Block diagram of the disparity calculation algorithm

The reason for having local search ranges (per line) is to be able to adaptively reduce the matching space. As already described in Section 3.5, the disparity estimator only searches along single horizontal image lines, so an adaptive calculation of the matching space per line is feasible. If a global range were calculated, a small object in the foreground with large disparity, would force a large search range for the entire image. This can be avoided using this local criteria.

Applying this limitation process to the "Aqua" sequence reduces the disparity search range to a maximum of [-5, +10] and the deviation search range to a maximum of [0, 1]. This way the matching space could be reduced to a maximum of 720x16x2 nodes per line in the worst case.

3.7 Perspective on Motion Estimation

Dynamic Programming has proved to be a suitable optimisation method for disparity estimation. With the enhancements described in Section 3.5, it is possible to estimate disparity without knowledge of epipolar geometry. With the three-dimensional matching space introduced, even motion estimation might become feasible using dynamic programming. However, a straightforward extension of the algorithm for use in motion estimation is not possible. In this Section, the necessary changes to dynamic programming for motion estimation are discussed.

As has been discussed in Section 3.3, ordering is an important constraint in dynamic programming. This constraint means that the objects on the two matching raster lines are in the same order. The reason for this is the way estimation is performed, namely sequentially along a raster line. Therefore, if the order is changed, dynamic programming will give no matching results.

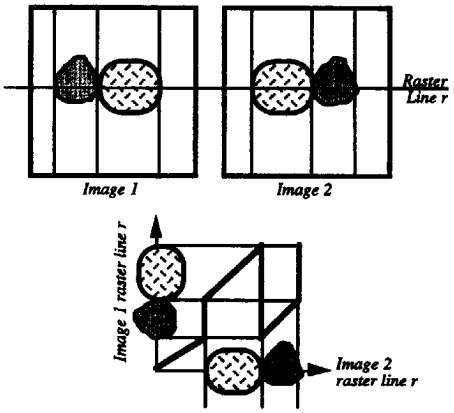


Figure 3.31: Ordering violation

Figure 3.31 shows an example, where the positions of the objects have been reversed in the two images. In this case, the ordering constraint is definitely violated. Although theoretically there exists a match as is shown in the lower part of Figure 3.31, dynamic programming cannot jump from one object to the next because the ordering constraint is violated. A possible solution to overcome this behaviour is to perform optimisation within the objects (intra) and not globally (inter) along the raster line. In Figure 3.31, although the ordering constraint is violated in terms of objects (their position is reversed) it is not violated within the objects. If large jumps are allowed at object boundaries but not within an object this problem can be solved. As the ordering constraint is not valid at such boundaries, the strategy for searching the predecessors in the dynamic programming also has to be changed. For a point (n,m) in the matching space, possible predecessors can now be in the entire column $n+1$, not just above row m, as is the case with disparity estimation (Figure 3.12). Such a possible path through the matching space is printed in bold in Figure 3.31.

As has already been described in Section 3.5, a *General usable Normalised Feature Difference* (GNFD) is used as the matching criterion. However, the matching window in this case should be designed to provide a large matching area with special strength in the vertical and horizontal directions. It is shown in Figure 3.32. For every possible motion vector (M_h, M_v) at point (n,m) in the search area, the *GNFD* is evaluated according to equation (3.13), as is done for disparity estimation.

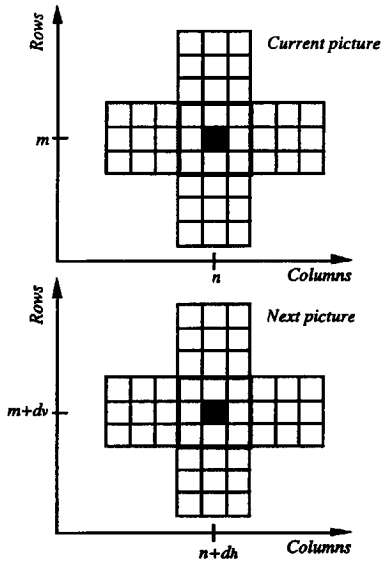


Figure 3.32: Window used for calculating the NFD at image point (n,m)

The critical aspect of using dynamic programming for motion estimation is the definition of the jump costs. On the one hand, vectors across a surface should be smooth and on the other jumps should be allowed at object edges. By reducing the effect of the jump cost at the edges, dynamic programming can use the matching cost *GNFD* as the main decision cost. Since matching near edges is usually better than within a surface, the *GNFD* is very reliable at edges leading to correct decisions without a high regularisation influence. A typical problem is when a large motion-vector jump occurs between two consecutive pixels belonging to two different objects along the raster line. This may be caused by the rapid motion of one of the objects. Therefore, the jump cost function of equation (3.15) will be extended, also taking the strength of a luminance edge into account.

The jump cost function $f(\delta)$ is defined as

$$f(\delta) = \frac{\left(0.3\sqrt{\delta} + 0.15\delta\right)}{s}, \quad \delta \geq 0, \; s > 0 \tag{3.22}$$

where δ is the magnitude of the motion vector jump between consecutive pixels along the raster line and s gives the strength of an edge through the gradient computed using the Sobel operator [Pra91]. The four different oriented sobel operators shown in Table 3.4 will be applied. The maximum value then is choosen as s. When this approach is adopted, $f(\delta)$ gives

higher values the "smoother" the image is, but gives very small values whenever an edge is present.

Horizontal Oriented Gradient	Vertical Oriented Gradient	45° Left Oriented Gradient	45° Right Oriented Gradient
$\frac{1}{4}\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix}$	$\frac{1}{4}\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix}$	$\frac{1}{4}\begin{bmatrix}-2 & -1 & 0\\ -1 & 0 & 1\\ 0 & 1 & 2\end{bmatrix}$	$\frac{1}{4}\begin{bmatrix}0 & -1 & -2\\ 1 & 0 & -1\\ 2 & 1 & 0\end{bmatrix}$

Table 3.4: The four different oriented Sobel operators

The accumulated cost value is then calculated using equation (3.17), so applying the same principle to motion estimation, as was previously explained for disparity estimation.

Experiments, performed with the "Tunnel" sequence shown in Figure 3.33, where only the train is moving,show that motion estimation using dynamic programming is indeed possible. The result of motion estimation performed in this way, i.e. depicting the horizontal and the vertical component separately, is shown in Figure 3.34 and Figure 3.35.

Figure 3.34 shows the horizontal component M_h of the motion vectors, where negative horizontal motion (from right to left) is shown in dark and positive horizontal motion (from left to right) is shown in bright colours. Most parts of the image do not have any motion, this is indicated by the grey colour. Figure 3.35 shows the vertical motion component M_v of the motion vectors. In the "Tunnel" sequence only negative vertical motion (from bottom to top) occurs, indicated by the dark colour.

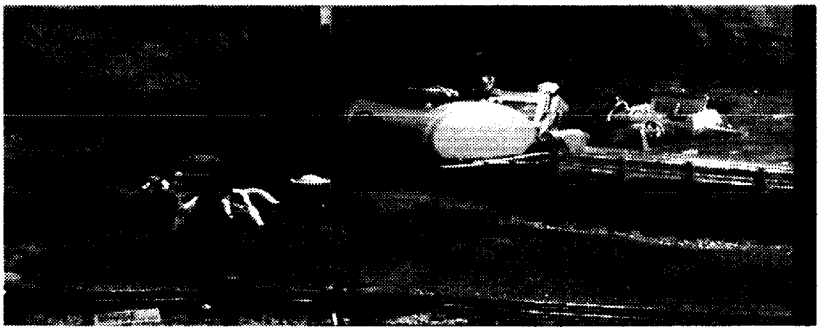


Figure 3.33: Field from the original "Tunnel" sequence

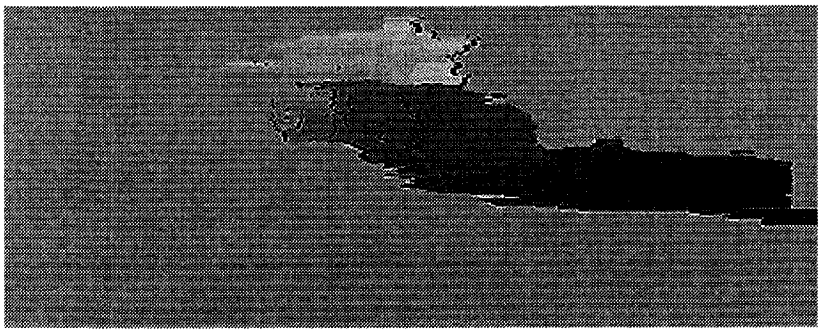


Figure 3.34: Horizontal component of the motion vector field

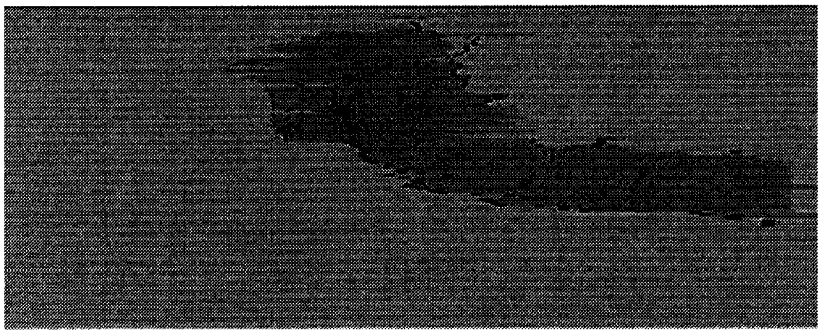


Figure 3.35: Vertical component of the motion vector field

3.8 Experimental Evaluation of the Disparity Estimation

In order to evaluate the quality of disparity estimation, it was tested with the "Mirror" sequence (Figure 3.17) as well as with the "Aqua" sequence (Figure 3.26). The advantage of the "Mirror" sequence is that the correct disparity vectors are known, as this is a synthetic sequence. In cases like these, the best evaluation criterion is the percentage of correct estimated vectors *(PCV)*. In the experiments carried out in this thesis this number only refers to the vectors, occluded areas will not be taken into account. In Figure 3.36 the percentage of correct vectors *PCV* per image is shown for the "Mirror" sequence. For comparison, the results of a full-search block matcher with block size 8x8, taken from [Fra96], are also included.

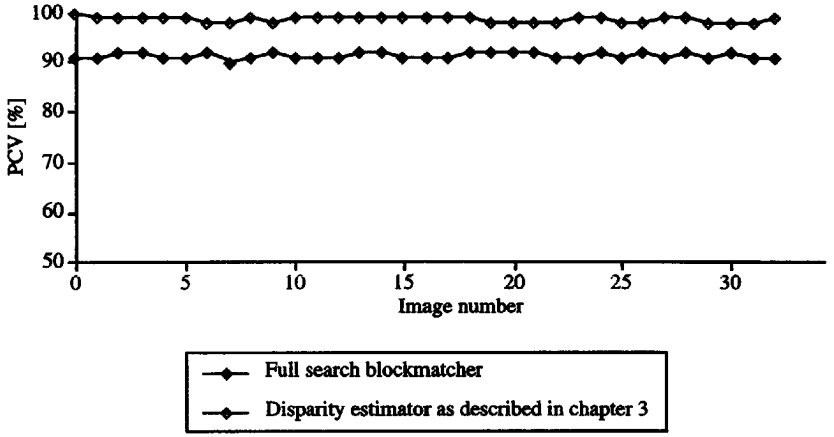


Figure 3.36: Percentage of correct vectors (PCV) for the "Mirror" sequence (30 dB noise)

In the first image (no disparity), 100% of the vectors are assigned the correct estimate, i.e. 0. In all the other images, around 98 to 99% of the vectors have been estimated correctly. Compared with a full-search blockmatcher, which estimates approximately 90% of the vectors correctly, this result is a lot better - as the graph shows.

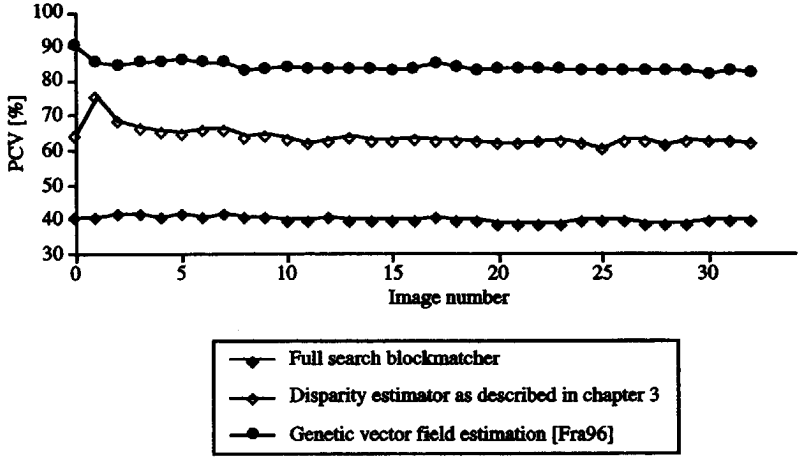


Figure 3.37: Percentage of correct vectors (PCV) for the "Mirror" sequence (20 dB noise)

Comparable results are obtained when disparity estimation is applied to the "Mirror" sequence with 20 dB noise, shown in Figure 3.37. It is not the focus of this thesis to estimate disparity in images having such a high level of noise, but the results in Figure 3.37 show that the vectors also improve in this case compared to a normal blockmatching. Compared to the genetic vector field estimator described in [Fra96] the results are a worse. Due to the high noise level, a lot of similar matches could be found, but dynamic programming does not allow the necessary disparity jump. The number of correctly estimated vectors is therefore a lot lower than with the 30 dB sequence.

With the "Aqua" sequence, evaluation of this kind is, unfortunately, not possible, as the true disparity values are not known. An approach to assessing the quality of the estimate is predicting the left image by shifting the pixel values in the right image with the corresponding disparity value according to equation (3.23).

$$W'_L(i,j) = W_R(i + D_h, j) \tag{3.23}$$

Whenever D_h did not exist for a pixel, but occlusion did instead, the luminance value of the corresponding pixel in the reconstructed image was set to zero. The quality of the reconstructed image W'_L was then compared with the original image W_L. The measurement used for this comparison is the Peak-Signal-to-Noise-Ratio *(PSNR)* defined as:

$$PSNR = 10 \cdot \log_{10} \cdot \frac{\pi \cdot \lambda \cdot 255^2}{\sum_{i=1}^{\pi} \sum_{j=1}^{\lambda} \left(W_L(i,j) - W'_L(i,j)\right)^2} \; [dB] \tag{3.24}$$

The result of this evaluation can been seen in Figure 3.38, where the PSNR-curves for "Mirror" and "Aqua" are printed.

With "Mirror", the reconstruction of the first image (disparity = 0) gives very high quality. As with larger disparities there are larger occlusions which cannot be reconstructed as there is no information available, the quality of the following images decreases rapidly. With disparities larger than approximately 15 pixels, the curve stabilises at a PSNR-value of about 19.5 dB. By contrast, the PSNR of "Aqua" stays approximately constant at just below 20 dB for all the sequence. In "Aqua", the amount of occlusion stays more or less the same in all the images, therefore only smaller PSNR deviations can be observed. Compared with "Mirror", it can be seen that the disparity estimation also works well on sequences like "Aqua", for example, where the epipolar line geometry is not known.

To provide a comparison, an experiment was also performed where the vertical deviations of the disparity vector field have not been allowed. The results are shown in Figure 3.39.

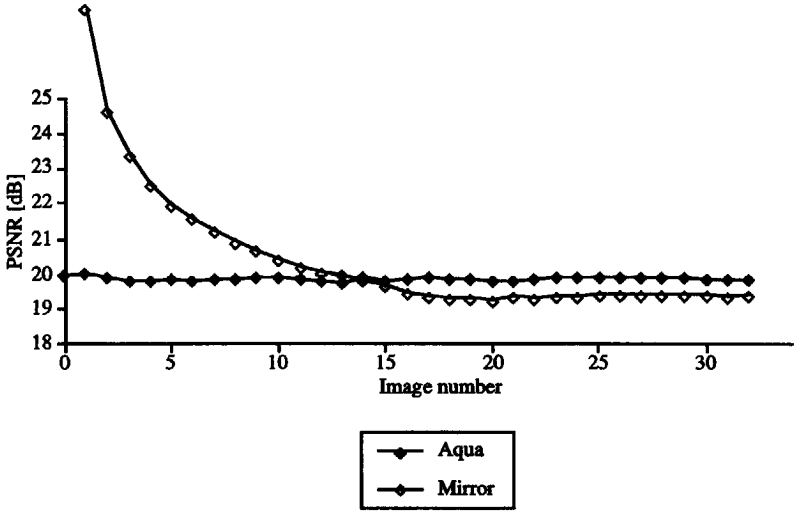


Figure 3.38: PSNR-curves for "Aqua" and "Mirror" after reconstruction of the left image from the right image and the disparity vector field

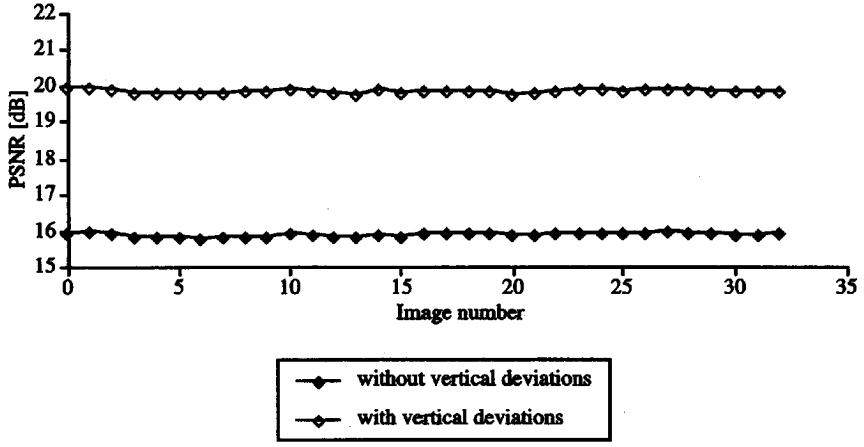


Figure 3.39: PSNR values of "Aqua" with and without vertical deviations in the dynamic programming approach

Here it becomes obvious that the introduction of vertical deviations is really important. An almost constant improvement of about 4 dB could be achieved, the vertical deviations also being taken into account.

3.9 Conclusion

In this Chapter, a new pixel accurate disparity estimator was discussed. Although the results of the experiments have been shown only for the "Mirror" and "Aqua" sequences, they are representative for a larger class of images, namely the DISTIMA test-sequences. Also for the other image sequences these results are fairly well reproducible. The disparity estimator can estimate pixel accurate disparity fields, even if the epipolar geometry is not known. It is based on a dynamic programming approach and takes the feature difference of the pixels into account, as well as a piecewise smooth disparity inside regions. Due to the optimisation approach being led through dynamic programming the estimator is also less sensitive to differences in foreshortening. The resulting disparity vector fields are accurate to within a pixel and are of high quality, as the evaluation showed. Therefore, they are a good basis for the further processing, the segmentation of regions being performed according to their disparity. The experiments applying dynamic programming also to motion estimation have shown that the estimation of a motion vector field also is possible with this approach. It therefore will be used not only to estimate disparity but also to estimate motion in the region-based coder described in the following Chapters. A couple of minor problems still remain, as there are occlusion gaps and line structures in the resulting disparity vector fields. These problems must be solved in a postprocessing step, before the vectors are forwarded to the segmentation process.

4. Image Analysis and Synthesis

Region-based analysis-synthesis coding - unlike block-based image coding - permits the description of arbitrarily shaped regions by means of certain parameters. As referred to already in Chapter 2, two very important parts of a region-based, analysis-synthesis coding system are *image analysis* and *image synthesis*.

At the encoder site, *image analysis* extracts the necessary parameters of the regions to be transmitted to the receiver site. These parameters depend on the source model used. Current implementations of region-based analysis-synthesis coders use either 2-D or 3-D models. The parameters used are motion, shape and colour of the region. In this Chapter, a new definition of the source model is developed. In addition to the "standard" parameters of a 2-D region, disparity information will now also be used to describe the region. As this representation also indicates the depth of a region it will be termed a $2\frac{1}{2}-D$ region. The use of such $2\frac{1}{2}-D$ regions will enable the system to encode stereoscopic sequences without a large overhead for the second channel and also general monoscopic sequences, assuming that disparity information is available, without *a priori* knowledge of the content.

At the receiver site, *image synthesis* uses the transmitted parameters to shift the regions and so synthesise the image. This, in principle, is a straightforward process: reconstructing the regions from the transmitted parameters and putting them in the correct place in the image to be synthesised. The introduction of disparity information will help resolve ambiguities which are a problem in monoscopic, region-based coders and occur if two regions overlap due to their motion. Image analysis and synthesis will not normally be capable of synthesising an image without error. Therefore, as a final step, the synthesis error must be transmitted to increase the quality of the resulting image. As the transmission of the synthesis error is neither part of the analysis nor of the synthesis, it will be discussed in the next Chapter.

Firstly, a definition of the source model used will be provided in the following. Then the overall concept of image analysis and synthesis will be described. For image analysis and synthesis, typical image processing tools will be used; they will also be discussed. With the aid of these tools, the next image in time and the spatially alternate stereoscopic image will be synthesised. Finally, the performance of the image analysis and synthesis will be evaluated by experiment.

4.1 The Source Model

The source model influences the image analysis part of a region-based analysis-synthesis coder in particular. In addition to, say, the 2-D regions used in [Höt90] - where a region is defined by its motion, shape and colour - $2\frac{1}{2}-D$ regions are also described by their disparity. Without loss of generality, only translational vectors for motion and disparity are used in this source model.

In the case of a monoscopic coding scheme, a very important factor is that of motion. In order to prevent an accumulation of motion errors, a region should have the same motion vector for all pixels in that region. In the case of a stereo-compensated coding scheme, disparity is the information needed per region. However, unlike to motion vectors, disparity vectors will be used only once to compensate the second stereoscopic image from the current, already motion-compensated image. Greater deviations than those encountered with motion compensation can, therefore, be allowed. These considerations lead to the following definition:

Definition: A $2\frac{1}{2}-D$ region is a solid cluster of connected pixels in an image, where each pixel in the cluster has the same motion vector and a disparity vector with a maximum deviation of 1 pixel from its neighbouring pixel.

From this definition, it follows that a pixel-wise motion and disparity estimation are required. An estimate based on dynamic programming as described in Chapter 3 provides a pixel-wise resolution of vector maps. Using disparity and motion vector maps of this kind, both having pixel accuracy, a pixel-based segmentation of the regions is possible.

When this source model is implemented, the defined regions will not correspond to what humans would consider to be an object - especially when only translational vectors are used. With the test sequences used in this thesis, it is impossible to segment a real physical object as one region as such regions have different disparity layers. Furthermore, if the real object rotates in the sequence, the use of translational vectors for segmentation will lead to a further partitioning of the object into several regions. In a coding environment as described in this thesis, this is not a problem as the regions will only be used to synthesise the next image in time or the second image of the stereoscopic image pair. Real 3-D modelling, and the possibility of segmenting and manipulating physical objects, would be preferable for an interactive system.

The source model consists of two sub-models: for the first image in a sequence or at scene-cuts, "translational moving *unknown* rigid $2\frac{1}{2}-D$ regions", for all the other images in a sequence "translational moving *known* rigid $2\frac{1}{2}-D$ regions" will be assumed. The reason for

this is simple to explain: since no *a priori* knowledge of the content is used, there is no knowledge about the regions of the scene in the first image. However, in the next images, the regions can be tracked from the previous image, assuming known regions. The only case where this will not work is if a new region - as e.g. uncovered background - occurs. This new region then has to be described using the source model of *unknown* regions again. For the second (stereoscopic) channel "translational displaced *known* rigid $2\frac{1}{2}-D$ regions" will be assumed whenever the regions are already known from the first channel. Occluded areas will be treated as *unknown* regions again. This leads to the fact that image analysis is necessary not only in the first image but also in subsequent images, although not the entire images will be analysed, but only small parts of them. Whenever the quality of the synthesised image falls below a certain threshold, the next image will be analysed completely. This principle is similar to that of introducing I-frames in MPEG. Whenever a local error occurs or the initial segmentation is not correct, this will lead to a synthesis error and the necessity to correct it as described later in Chapter 5.

4.2 Concept of Image Analysis and Image Synthesis

This Section gives an initial conceptional overview of the use of image analysis and image synthesis in the region-based stereoscopic coder for the purpose of synthesising an image. Detailed explanations will be given in the following Sections.

Image analysis (see Figure 4.1) is carried out using the motion vector field from images L_i to L_{i+1} and the disparity vector field from images L_i to R_i. With images L_1, L_2 and R_1, image analysis is performed for the first time, so segmenting the entire image into regions. With the succeeding images, segmentation is restricted to image parts, where no information is available at the moment. These image parts (areas of uncovered background and occlusion shown in black in Figure 4.1, detected as areas without any information after the synthesis step) will be used as separate regions for the synthesis of the current images. Merging of such newly segmented regions with already existing regions will be investigated when motion and disparity information is available for them in the next time instance.

Since the source model that has been used is based on rigid regions, the regions' parameters will not be updated. It will, therefore, be necessary to transmit the parameters for every new region: in the first image of a sequence when the regions are defined for the first time and, in the subsequent images, for uncovered background, occlusions and for merged regions.

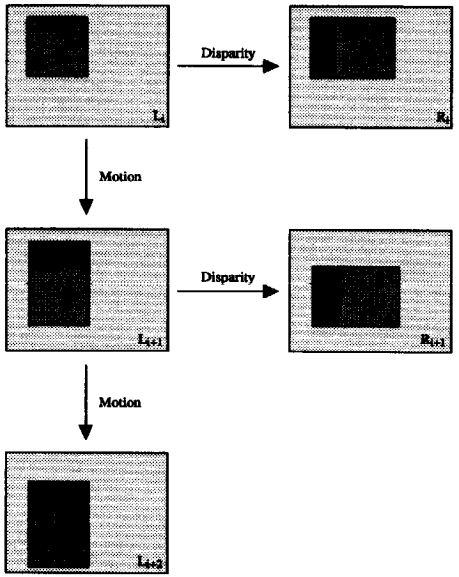


Figure 4.1: Image analysis based on motion and disparity vector fields

The basic principle of image synthesis is shown in Figure 4.2. The objective of image synthesis is to synthesise an image with good visual quality and so make it unnecessary to send an error signal. Image synthesis is therefore different from prediction as performed in block-based coders. The left images will be synthesised by shifting the regions according to their motion vectors. Based on this, the right images will be synthesised by shifting the regions according to their disparity vector as well. For each region, there is a region memory (to be described later) which contains the parameters colour, shape, motion and disparity. By summing the relevant displacement vectors, the new position of the region in images L_{t+k} and R_{t+k} is calculated. For the first, left image in a sequence, image synthesis assumes zero motion. This means that the regions have the same position as they did in the original left image.

The principle of image synthesis assumes that a motion estimator is able to follow the regions over time. A region-based motion estimator [ZP93], therefore, tries to match the known regions of the region memory to areas in the next image. Based on a histogram of the pixel-accurate motion vectors estimated as described in Section 3.7, the vector which appears most often in the region is chosen as the representative motion vector for the entire region. As long as the region of interest is not covered by another region, this works well. Even minor overlaps can be handled in this way. The image area will still be identified as the

corresponding region. If the overlap is too large, a match cannot be found. The region then cannot be motion compensated and a synthesis error occurs. When the region becomes visible again in later images, there is a good chance of identifying it again as a region from the region memory. If this is not possible, it will be added to the memory as a new region.

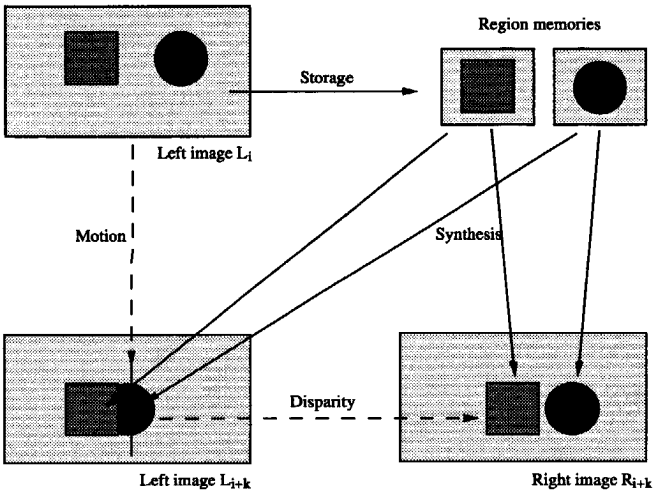


Figure 4.2: Image synthesis using motion and disparity, based on a region memory

4.3 Image Analysis

If the source model described in Section 4.1 is used, the image analysis part will have to define the regions and extract their parameters for further processing. As these regions will be used for coding, the following requirements and considerations must be fulfilled:

- As a consequence of the definition of a $2\frac{1}{2}-D$ region and the source model used, the pixels assembled to form a single region must have the same motion vector and a similar disparity vector.

- All the parameters for each region have to be transmitted. These regions may not be too small. For very small regions it might be better - from the coding point of view, which is the number of bits necessary for the description - to merge them with a neighbouring region, so taking a coding error into account.

- Generally, there always has to be a compromise between the number of bits required to transmit the region (including all its parameters) and the consequences of not transmitting the region, in terms of the number of bits and image quality.

Concerning the last two points, several publications deal with the problem of looking for an optimal solution in the rate-distortion sense. In [SSG96] a rate-distortion criterion is used to decide which regions should be used for compensation and for optimising the contour representation. In this thesis, in contrast to the latter and others (e.g. [MMP96] where an optimal segmentation is based on the rate-distortion theory), a simpler, non rate-distortion theory based decision criterion will be introduced. This criterion reduces the calculation effort (compared with a rate-distortion approach) but nevertheless achieves high-quality results.

Taking the above considerations into account, the image analysis part will execute the following steps (Figure 4.3):

1. Postprocessing of disparity maps
2. Region segmentation
3. Extraction of region parameters
4. Merging of small regions with larger neighbouring regions
5. Merging of equivalent regions
6. Description of the region shape

Postprocessing of disparity maps first carries out an interpolation of the occlusion gaps and removal of the line structures introduced by the dynamic programming - as was seen in Chapter 3 - which disturb the segmentation process. *Region segmentation* then defines a set of $2\frac{1}{2}-D$ regions. Each $2\frac{1}{2}-D$ region is subsequently processed to extract properties (*Extraction of region parameters*) such as statistical information, neighbourhood information and other parameters required for further use. These parameters are used in a process *Merging of small regions with larger neighbouring regions*, which, if certain criteria are satisfied, merges a small region with a larger one to minimise the total number of regions. These parameters are also used for *Merging of equivalent regions*, which is performed if new regions are added to the existing regions in the memory. It will check whether it is possible to merge the new region with an existing one or whether it has to be treated as a single region on its own. Both merging steps are done iteratively until all possible regions are merged. *Description of the shape region* concludes the analysis part - describing the final shape of the region and performing polygon approximation.

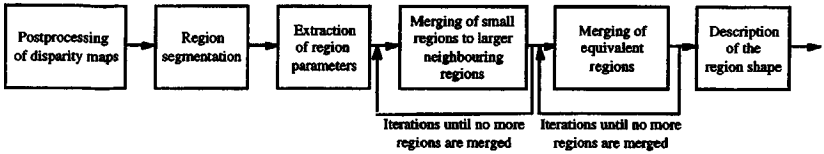


Figure 4.3: Image analysis

In the case of first image, all the regions will be "initialised", but in the case of the subsequent images, existing regions will be shifted according to their motion. In order to guarantee temporal consistency of a region, a region once segmented will not be changed. However, when there is a new region - as e.g. uncovered background or areas of model failure - this will be examined and possibly merged with an existing neighbouring region, or, if the criteria for merging are not fulfilled, it will be treated as a new region on its own. These new regions will occur whenever background in the image is uncovered in one of the subsequent images or whenever there is model failure. These areas will be treated as candidates for new regions. Based on the segmentation process, a decision will be made to keep them as a separate region or to merge them with an existing one. This merging (either performed in *Merging of small regions with larger neighbouring regions* or in *Merging of equivalent regions)* is the only way to change an existing region.

4.3.1 Postprocessing Disparity Maps

As the described disparity estimation method (using dynamic programming) functions independently on single lines, there will always be some line artefacts, as shown in Section 3.5.4. These line artefacts would not be a problem with pixel-wise stereo-compensation. However, as the vector fields are used for segmentation, these artefacts have to be removed to obtain smooth region boundaries. Smooth region boundaries are easier to describe and so do not need as many bits as coarse boundaries. *Postprocessing of disparity maps* will do so by filtering the line artefacts. The described disparity estimation method also includes occluded areas in the resulting disparity map. Although these areas are only visible in one of the two stereoscopic images, they belong to a region and a decision has to be made as to which region it should be assigned. *Postprocessing of disparity maps* is also a preparation for this decision, as disparity values are introduced at occluded areas by means of interpolation. In this way, a dense vector field without any gaps and artefacts can be input into the segmentation process and the disparity map can now be completely segmented.

Post-processing of the disparity map has three phases:

1. Median filtering with template size 3×3 pixel
2. Interpolation of occlusion gaps
3. Median filtering with template 1×5 pixel

Since the only objective of this postprocessing step is the removal of line structures and the filling of occlusion gaps, it is not actually a particularly crucial decision what filter to use. For the sake of simplicity, simple median filters were used. Also other types or sizes of filter could be used without seriously influencing the result.

The 3x3 window median filter is used to adjust each pixel to its neighbourhood. A median filter is known to respect edges, so it was selected so as not to disturb the disparity edges, which are very important in segmentation. Figure 4.4 shows the result after applying this filter to the disparity vector field of "Aqua" shown in Figure 3.28.

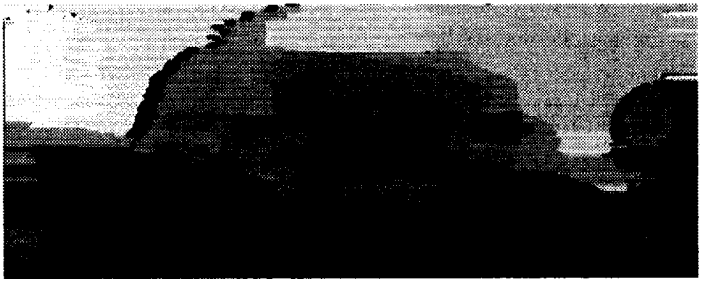


Figure 4.4: 3x3 median filtering of disparity vector field

The second phase of filtering aims to fill up the occluded areas. Assuming that the disparity estimation can precisely detect areas of occlusions, one simple way of interpolating occlusions would be to extend the region in the background - where the occlusion will occur - until it reaches the region in the foreground. As has been shown in Chapter 3, this assumption is not true in all cases. With dynamic programming there will be some incorrect estimates near occlusions - the horizontal line structures show that. As these line structures will be removed by simply comparing adjacent lines, it cannot be guaranteed that all the occlusions will be detected correctly. Therefore, the simple method described above - extending the region in the background - may be unsuccessful.

To avoid such problems, the filling process is performed by using the gradient information of the original luminance signal. The top graph in Figure 4.5 shows the gradient along a horizontal raster line of the image extracted from luminance. The same section of the horizontal line extracted from the disparity graph is depicted in the middle graph of

Figure 4.5. As occlusion will only be recognised when there is a disparity jump, occlusion indicates a region boundary with a depth difference with respect to the neighbouring region. Assuming that the region boundary will be visible as an edge in the image, the disparity values have to change at the region boundary or the peak gradient.

From the two graphs at the top of Figure 4.5, one can see that there is an occluded area on either side of the peak gradient which is defined as the dividing line between the two regions. The filling process constructs a gradient curve (Sobel operator) beginning from the point where the occlusion starts and ending where it stops. To the left of the peak gradient, which is the maximum value of the gradient in the occluded area, the gap is filled with the disparity encountered before the occlusion; to the right of the peak gradient, the gap is filled with the disparity encountered after the occlusion as indicated in the bottom graph of Figure 4.5. The whole process is performed independently on lines and repeated for all the occluded areas in the image.

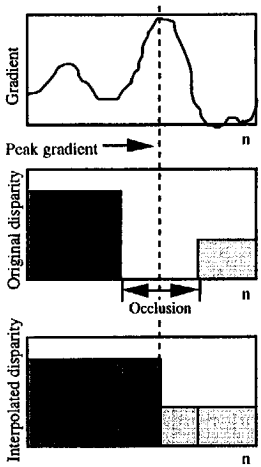


Figure 4.5: Illustration of interpolation of an occlusion gap

The interpolation of occlusion gaps is a critical part of post-processing. If inconsistencies are present in this process, this leads to an inaccurate definition of regions in the segmentation process later on.

As the described operation will not result in a connected edge, but in peak gradients not connected to one other, the final result of the interpolation process will not have smooth contours but coarse ones. Figure 4.6. shows the result of the interpolation process on all of the image.

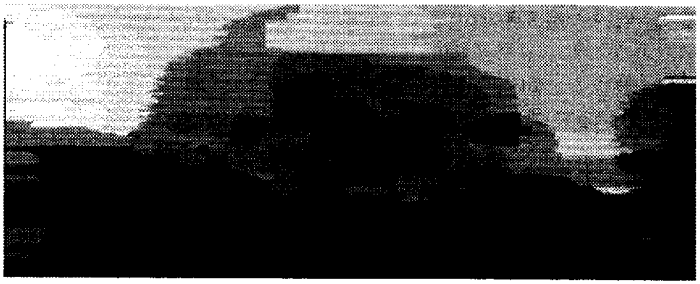


Figure 4.6: Interpolation of disparity vector field

In some cases, this interpolation will produce the same result as the simple background-foreground interpolation referred to at the beginning. The result will be the same whenever both the gradient and the occlusions are detected precisely, which is not very often the case. With occlusions in particular, there will be some problems as has been stated before. On the other hand, using the gradient method, there is a danger that this might lead to coarse edges, as the peak gradient might differ from line to line. Because of this possible effect and the line structures from the dynamic programming, a final simple step is vertical filtering to eliminate these disturbing points. A simple approach with sufficiently high quality is a median filter with a template size of 1x5. Figure 4.7, therefore, shows the final disparity vector field of "Aqua" which is used for further processing. The remaining artefacts - there still are some coarse edges visible in the image - can be handled and so are acceptable for further processing.

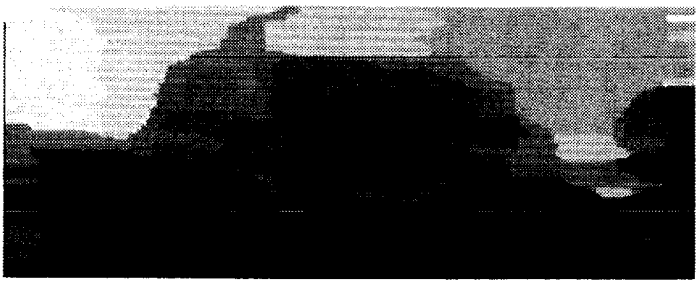


Figure 4.7: Vertical median filtering of disparity vector field with size 1x5

The need for postprocessing can also be illustrated experimentally. Using the disparity map, the right image was compensated from the left one. The results were compared with the original image to determine the Peak Signal to Noise Ratio (PSNR).

Table 4.1 shows that each postprocessing step improves the quality of the vector map.

Disparity Map	PSNR
Raw (Figure 3.28)	19.9
Median filtered (Figure 4.4)	20.8
Interpolated (Figure 4.6)	24.9
Vertical filtered (Figure 4.7)	25.0

Table 4.1: The usefulness of post-processing in correct correspondence calculation

As the PSNR values are calculated after stereo-compensation of the left image, the interpolation step makes the largest gain. As there will be no information in occluded areas, they will not have any information in the stereo-compensated image, so reducing the quality of the image. After interpolation, there are no occlusions left and so grey values will be present in all of the image.

4.3.2. Segmentation of Regions

Many different segmentation methods (e.g. [Wal91] and [SSJ95]) can be found in the literature. What they all have in common are the following two ideas [Nie90]:

- *The result of a segmentation is a set of regions, each region having certain attributes.* With image segmentation, the most important types of segmentation regions are line segments, lines, regions and volumes. Possible types of attributes include the location in two-dimensional image co-ordinates or in three-dimensional world co-ordinates, colour, texture, motion, depth, shape, etc. Apparently not every type of attribute is adequate for every type of region.

- *Segmentation requires some changes or fluctuations of suitable pattern properties.* Segmentation of a pattern represented by a function $p(n, m)$ is obviously impossible if this function is just a constant. To a human, a constant function may invoke some sensory impressions, but it is evidently without structure. Therefore, an idea common to all techniques of pattern segmentation is that regions are related to changes in $p(n, m)$. Changes in $p(n, m)$ suggest possible borders between these regions.

On the basis of the method used for image synthesis, two requirements have to be fulfilled:

- The errors in motion have to be kept to a minimum and so a region must have a motion vector which is identical for all pixels in that region - for the horizontal part of the vector as well as for the vertical part.

- With disparity vectors, larger errors than those occurring with motion compensation can be allowed. In contrast to motion vectors, which will be summed, disparity vectors will only be used once. Therefore, an accumulation of errors is avoided with disparity. This leads to the concept that not all the pixels within a region have to have the same disparity.

As texture and colour information are already implicitly used in the vector estimation itself and especially in the interpolation of the occlusion gaps, it will not be used again for segmentation. This would make the system more complex without using new information for the segmentation process. In [Kir89] the segmentation of vector maps has been proven to be quite robust, it was, therefore, also selected for this system.

The segmentation of the displacement vector field

$$I = \left\{ \left(\vec{D}_{n,m}, \vec{M}_{n,m} \right); n = 1,\ldots,\pi; m = 1,\ldots,\lambda \right\} \tag{4.1}$$

into connected regions $\Re_i$

$$I \rightarrow \left\{ \Re_i | i = 1,\ldots,\omega \right\} \tag{4.2}$$

is performed so that

$$I = \bigcup_{i=1}^{\omega} \Re_i \quad \text{and} \quad \Re_i \cap \Re_j = \varnothing \quad \text{for} \quad i \neq j \tag{4.3}$$

The fundamental aspect of a region is *homogeneity*. A simple region growing algorithm [Pra91] (based on a 8-connective neighbourhood definition) is used to find the connective regions which satisfy the criterion

$$\left(\left| D_{h_{i,j}} - D_{h_{k,l}} \right| \leq \Theta_d \right) \wedge \left(\left| M_{h_{i,j}} - M_{h_{k,l}} \right| \leq \Theta_h \right) \wedge \left(\left| M_{v_{i,j}} - M_{v_{k,l}} \right| \leq \Theta_v \right) \tag{4.4}$$

for all $(i,j),(k,l) \in \Re_i$

where $\Theta_d, \Theta_h, \Theta_v$ are threshold values for either disparity, horizontal or vertical motion selected as $\Theta_d = 1$, $\Theta_h = \Theta_v = 0$. In contrast to, say, [MB94] and [DC95], where an affine model is used to segment the regions, only translational vectors are used in this case, corresponding to the source model that is employed.

The segmentation is controlled by a mask which determines the areas of the image that can be segmented. Therefore, an area that has already been segmented is no longer marked, so avoiding a further segmentation. On this mask, therefore, the occluded areas and the areas of uncovered background are marked. These are extracted by forward image reconstruction using the regions from the previous image and their motion and disparity vector. In this way, uncovered background and occlusions become apparent as areas with no information at all in the synthesised image and it is now possible to segment these regions. For the first image, the mask shows that the entire image is segmentable. The way the segmentation is performed on the vector fields is tracing their homogenous parts through region growing [Pra91]. The set of parameters D_h, M_h, M_v is evaluated and if their parameter values do not violate the homogeneity criterion, this is taken as evidence that no transition to another region occurred.

Using this procedure, it is guaranteed that connective regions satisfying the homogeneity criterion will be found. Of course, with the presence of inconsistencies in the vector field, these regions might be very small and will not have any physical meaning as has already been explained in Section 4.1. Figure 4.8 shows the result of the initial segmentation procedure for the "Aqua" sequence. The whole vector fields are segmented for the first image.

Figure 4.8: Initial segmentation of first image of "Aqua" containing 669 regions

This result is based on the disparity vector field shown in Figure 4.7 and the motion vector field calculated using equation (4.4) (not shown here, as there is almost no motion in the "Aqua" sequence). The lack of motion gives a segmentation that, in this example, is almost entirely based on the disparity field alone. A different colour value is assigned to neighbouring regions for all the 669 regions defined in the segmentation process.

4.3.3 Extraction of Region Parameters

After the image has been segmented into regions, the following parameters need to be extracted for each region for subsequent processing:

- a region identification number i,
- the number of pixels A_i within the region,
- the motion vector $\vec{M}_i$,
- a list of neighbouring regions $\Re_j, j = 1,...,\omega$
- the number of common pixels on the boundary of the neighbouring region $B_{i,j}$
- the disparity D_k.

Most of these parameters can be calculated in a straightforward manner, but the last one - disparity - requires more attention. The reason for this is that, according to the requirements of segmentation, there will normally be a number of different disparity values within a region. For later use in the region-based stereo-compensated scheme, only one disparity vector can be used per region. The selection and assignment of a single disparity vector is a non-trivial problem.

It is easy to think of some straightforward algorithms to determine which disparity vector should be taken as representative:
- calculate the mean disparity of all present disparity values for the region,
- calculate a median disparity vector or
- take the vector which is correct for most of the pixels.

In this region-based coding environment, the mean square error (MSE) is used as the criterion for assessing the quality of the synthesised image. In this sense, all of the methods mentioned above will give suboptimal results as far as the synthesis error per region is concerned because they do not take the MSE into account. The only way of finding the optimal representative vector for a region is to compensate that specific region with all the possible disparity values and to calculate the MSE as a measure of quality. The disparity value achieving the best quality for the whole region will then be taken as the only disparity value for the region. Therefore, the disparity vector with a minimum MSE for the region will be selected as the representative disparity vector for the entire region.

As required by the source model, known translational moving regions will be assumed, starting with the second image in a sequence. As these regions may change over time, due to merging with other neighbouring regions, the parameters might change also. An update of the parameters will always occur if the region changes.

4.3.4 Merging of Small Regions with Larger Neighbouring Regions

In a region-based coding system, the region shape must also be transmitted. However, when a region is very small, it may be that the shape information takes more bits than would be required if the region were not transferred and the coding error sent instead. For that reason, a compromise has to made for small regions of this kind. There are actually two possibilities: either to transmit the small region on its own or to merge it with a large neighbouring region.

The objective of merging small regions with a larger neighbouring region is to obtain a single short contour, while keeping the synthesis error at a low level, after compensation with motion or disparity vectors. Approaches such as merging with the region which yields the smallest possible synthesis error would be better for one type of compensation only - either motion or stereo - but would not take the other into account. As the final regions will be used for both motion and stereo compensation, another approach was developed.

A quantitative decision whether to merge the small region - and if so, with which neighbouring region - would require a lot of calculations as all possibilities would have to be calculated and compared. What is more, taking into account the fact that the boundary of a region need only be transmitted once, whereas the corresponding synthesis error would also occur with the subsequent images, error propagation would also have to be considered. The decision, therefore, will be based on an estimate of the effects. The obvious way to estimate which solution is better is based on the size of the region, comparing the probable bit rate of the shape with the probable bit rate of transmitting the synthesis error.

First, the selection of all possible candidate regions $\Re_j$ with which a small region $\Re_i$ could be merged. For the neighbour to qualify as a candidate, the following must apply:

$$A_i \leq A_j \tag{4.5}$$

For all the qualifying neighbours, equation (4.6) will be evaluated. This criterion takes into account the difference of disparity, horizontal motion and vertical motion, but also the number of pixels sharing the boundary $B_{i,j}$. The neighbour for which the quality criterion $QS_{i,j}$ is minimal is selected as the appropriate one for merging with region $\Re_i$.

$$QS_{i,j} = \left(\frac{\left|D_{h_i} - D_{h_j}\right| + \left|M_{h_i} - M_{h_j}\right| + \left|M_{v_i} - M_{v_j}\right|}{B_{i,j}^2} \right) \tag{4.6}$$

The criterion is designed to select the neighbour $\Re_j$ so that two neighbouring regions have:

$$\left|D_{h_i} - D_{h_j}\right| \Rightarrow \textit{a small absolute disparity difference}$$

$$\left|M_{h_i} - M_{h_j}\right| + \left|M_{v_i} - M_{v_j}\right| \Rightarrow \textit{a small absolute motion difference}$$

$$B_{i,j}^2 \Rightarrow \textit{a large common boundary}$$

The merging process which has been developed is iterative - it starts with the smallest region - and is repeated until no more small regions can be merged. In this way, all small regions will be merged.

The question of the maximum size of a small region has remained unanswered until now. It is necessary to define a threshold so that up to size A_s a region $\Re_i$ is considered to be a *small* region. As with the probable bit rate for the contour, the necessary number of bits for a highly sophisticated contour coding method is estimated at 0.3 bits per contour point in [Höt90a] - assuming a maximum contour error of 3 pixels and an average length of 2000 pixels per contour. A region with a size of, say, 100 pixels will have 28 contour points in the best case - assuming a "digital" circle - and up to 100 contour points in the worst case - assuming a region where no points are part of the region except the contour points themselves, e.g. in a 2x50 pixel block. These estimates assume that a region has a least width of 2 because the estimated contour will be part of the region, as will be shown later. Therefore, between 8 (28 contour points times 0.3 bits per point) and 30 (100 contour points times 0.3 bits per point) bits for the description of the region contour will be needed. As regions in this system will have an arbitrary shape, the estimated number of bits required to describe the shape should be somewhere between these extremes - about 15 to 20 bits on average.

The estimate for the coding error is based on an average bit rate of 0.2 bits per pixel, which can be obtained with standard algorithms, e.g. MPEG2 and others [RH96]. For a region with 100 pixels this gives an estimate of 20 bits for the coding error.

From the above estimates, it can be seen that, for regions smaller than 100 pixels, transmitting an additional error will probably be cheaper in terms of bits than transmitting the shape information. The effect is even more marked if the subsequent images are taken into account as the error would have to be transmitted again. For regions with more than 100 pixels, the opposite applies. With these numbers, the strategy is clear: regions with a size A_i less than 100 pixels will be merged with one of the neighbouring regions. However, seen in relation to all the other bits that are required, these few bits will not unduly affect the final result from the region-based coder. The decision about the size of the regions to be merged is, therefore, not a very critical one.

4.3.5 Merging of Equivalent Regions

When uncovered background or occluded areas become visible in the scene, new regions will be created in addition to the existing ones. It will then be necessary to check whether these new regions should be treated as new single regions or whether they should be merged with existing regions. Merging of equivalent regions will take place in all the images of a sequence. Even in the first image - after the initial segmentation of the image into regions - this step will help to reduce the number of regions.

This merging is a two-stage process. The first step finds suitable neighbours as candidates for merging. If there is more than one suitable neighbour, a second stage uses a certain criterion to find the "best" neighbour.

Only two regions, $\Re_i$ and its neighbour $\Re_j$, can qualify for merging if equation (4.4) is satisfied. For all the qualifying neighbouring regions, the following quality criterion $QE_{i,j}$ will be used:

$$QE_{i,j} = \frac{(A_i + A_j)}{B_{i,j}^2} \cdot (\bar{I}_i - \bar{I}_j)(\sigma_i^2 - \sigma_j^2) \tag{4.7}$$

where the criterion is designed to choose the neighbour $\Re_i$ so that the two neighbouring regions have:

$B_{i,j}^2 \Rightarrow$ *a large common boundary*

$(A_i + A_j) \Rightarrow$ *a large merged area*

$(\bar{I}_i - \bar{I}_j) \Rightarrow$ *a small luminance mean value difference*

$(\sigma_i^2 - \sigma_j^2) \Rightarrow$ *a small luminance variance difference*

The objective of using criterion (4.7) is to select the best possible merging. The major argument for selecting a certain region is the luminance difference between the two regions, measured in terms of the luminance mean value of the complete region and its variance. If this argument is not unambiguous, the number of common boundary points and the area of the regions will influence the result. This will then lead to large regions which have a large common boundary.

Merging of equivalent regions is again an iterative process - starting with the smallest region - and is repeated until no more regions can be merged. Figure 4.9 shows the segmented regions after the two merging processes, starting with the region map shown in Figure 4.8.

504 regions of the 669 regions that were initially segmented have been merged as small regions; another 40 equivalent regions have been merged. Figure 4.9 still shows 125 regions.

Figure 4.9: Region map of "Aqua" with 125 regions after merging of small and equivalent regions

4.3.6 Description of the Region Shape

The description of a region shape is a well-known problem. The basic ideas are either to obtain a hierarchical description of the region [PN95, CL95] or to describe the region using triangular nets [Sch95]. To achieve this, several approaches - such as using a moment difference method [YML95] or morphological skeletons [BK95] - can be used. The objective here is not to describe the shape precisely, which, for example, could be effected by run-length coding of all the contour points, but to approximate the shape within a certain permissible error. The regions used in this thesis are assumed to be solid, so only the outer boundary of the region will be approximated.

Methods using either Fourier descriptors [PF77] or polygon approximation [Höt90a, RG96, SIM94, WZJ96] have been shown to be very efficient. Both of the two methods mentioned have similar efficiencies [Fra89]. However, when Fourier descriptors are used in areas with strong curves, a large number of coefficients are required to describe the contour. It also is very difficult to use the information from preceding images to predict the current shape parameters [Höt90].

Polygon approximation does not have these disadvantages. Describing strong curves, or even corners, poses no problems and it is possible to use the shape of the preceding region efficiently to predict the current parameters. The only disadvantage of polygon approximation is that a shape described in this way does not look very natural but angular. With regions that correspond to natural regions as humans would define them, this might be a problem. In [Höt90a], for example, the goal was to approximate the shape of a person which was defined as one region. Therefore, a combination of splines and polygons was used in that paper to overcome this restriction.

With the regions defined here - which have nothing in common with "natural" regions - a description using only polygons is sufficient. The maximum permitted error for polygon approximation has to be defined. The coding scheme that has been developed will be able to quite easily handle regions whose size has been overapproximated. This is because the disparity information will help to solve ambiguities when two regions overlap. Based on the results in [Höt90], a maximal deviation of 2 pixels outside the region is defined for this case.

In contrast to deviations outside a region, deviations within a region are a source of large synthesis errors, as a collection of regions that are too small will leave areas in between these regions without information after synthesis. To minimise these effects, no errors within the region have been allowed. The maximum error of the contour description inside a region is, therefore, set to 0 pixels.

To keep the contour as small as possible, - so as to reduce the number of bits required for the description as much as possible - the contour was selected so that it belonged to the region. As a result, there are regions with a minimum width of 2 pixels. The method used for the description of the region shape has two steps [SIM94]. The first to obtain the main vertices of the contour and the second an iterative polygon approximation.

Proceeding clockwise along the region boundary, the distance between all possible pairs of points on the contour is calculated. This calculation is based on the 8-connective neighbourhood definition. The pair of points with the maximum separation is used as two initial vertices for the approximation (vertices 1 and 2 in Figure 4.10).

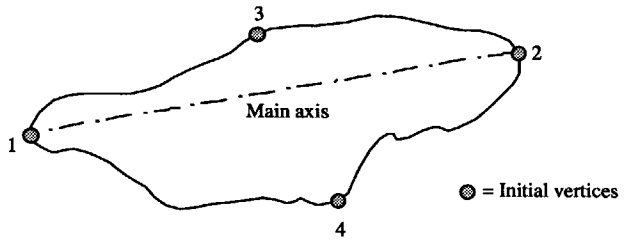


Figure 4.10: Initial vertices

The axis between them is referred to as the main axis of the region. Proceeding clockwise along the contour from the first main-axis vertex to the second main-axis vertex, the perpendicular distance from this point to the main axis is calculated for each point on the contour. The point with the largest separation is called a vertex of the approximation. This is repeated for the anti-clockwise direction. The two new initial vertices are numbers 3 and 4 in Figure 4.10. After the definition of the initial vertices the polygon approximation starts. First, a straight line is drawn between the two initial vertices 1 and 4 from Figure 4.10. For each

point on the contour between these two vertices, the distance between the point and the straight line is calculated.

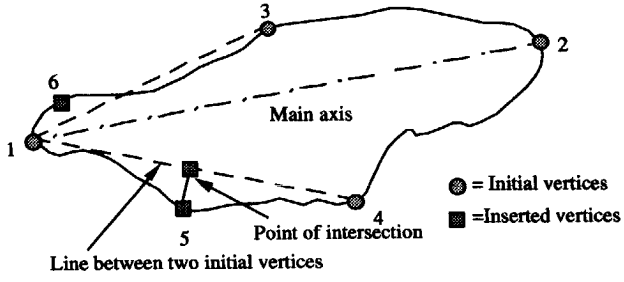


Figure 4.11: Polygon approximation

If the separation for a point is less than or equal to the threshold of 2 pixels for a point outside the region or 0 pixels for a point within the region, no new vertices need to be inserted and the analysis can continue with the next pair of vertices. However, if the separation exceeds the threshold, a new vertex has to be inserted. The new vertex is inserted at the contour point which has the maximum separation from the line between the two initial vertices (Figure 4.11). The process is repeated iteratively until no more vertices can be inserted (Figure 4.12).

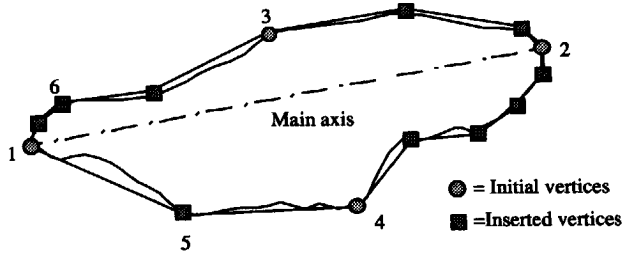


Figure 4.12: Completed polygon approximation

Figure 4.13 shows the polygon-approximated regions - with the regions shown in Figure 4.9 - of "Aqua" superimposed on the original image.

When a polygon approximation of this kind is used, each region will have its own contour. Therefore, double contours can be seen in Figure 4.13 when two regions overlap each other by more than one pixel or the two regions are segmented without any overlap. The only situation without overlapping contours is that of two regions overlapping each other by exactly one pixel - the contour itself. With the disparity-based concept introduced in Section 4.2, this is not a problem as it is always known which region is visible and which region is covered.

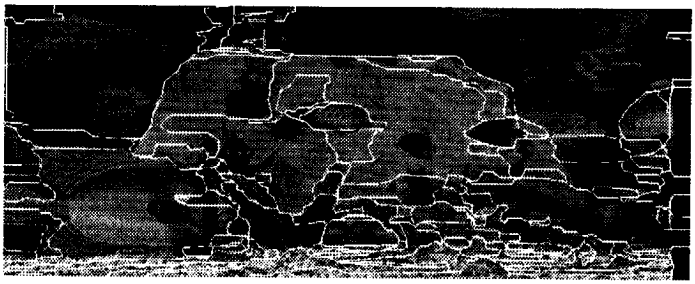


Figure 4.13: Polygon-approximated region-contours superimposed on original image

4.4. Image Synthesis

Image synthesis naturally takes place at the receiver side to synthesise an image, but synthesis is also necessary at the sender side to allow the image in question to be synthesised at the receiver. Using the synthesised image, the sender can check changes in the parameters and so be able to calculate, say, motion and disparity based on the same image as the receiver and avoid error accumulation.

There are two different ways of performing image synthesis with this kind of coder [Höt92]:

- based on an image memory, where the last coded and decoded image is stored (familiar from the standard block-based coders) leading to a synthesis based on the previous image or
- based on a region memory, where all the regions and their parameters are stored in a central memory.

Both methods will now be described in greater detail under the assumption that the images can be synthesised using only motion and disparity information. In the stereoscopic system described, the left images of the stereoscopic sequences will be synthesised using the regions' motion vectors. To reconstruct the right image of the stereo pair, the same regions will simply be shifted according to their disparity vector as well.

As will be described in the next sections, the two methods under discussion - one based on an image memory and the other based on a region memory - differ from each other in two respects:

- their behaviour as regards subpixel accurate vectors;
- their behaviour as regards the stored information when new regions are added to the memory.

Although only pixel accurate vectors are used in the system that has been developed, a further improvement in the system will make it necessary to take subpixel accuracy of the vectors into consideration as well. For this reason, the following discussion assumes that subpixel-accurate displacement vectors can be used, so allowing a further improvement in the system without changing the synthesis part.

4.4.1 Image Synthesis based on an Image Memory

The principle of image synthesis based on an image memory is shown in Figure 4.14 for a one-dimensional situation. In this example with three temporally successive images, the colour values of the image *k+1* to be synthesised will be taken from the image memory, which is the previously synthesised image. This image memory holds the synthesised image *k*. The motion vector field $\vec{M}(n,m,k+1)$ describes the displacement of image *k* with respect to image *k+1*. In Figure 4.14, each vector of $\vec{M}(n,m,k+1)$ assigns a position *(m, m, k)* of image *k* to a position *(n, m, k+1)* of image *k+1*. For the image synthesis of image *k+1*, the values of image *k* will then be taken and inserted at the displaced positions. If the motion vectors can have subpixel accuracy, the colour values will have to be interpolated from the surrounding values. In Figure 4.14, for example, the colour values at *(n, m, k+1)* and *(n, m+1, k+1)* have to be interpolated from the colour values at *(n, m, k)*, *(n, m+1, k)* and *(n, m+2, k)*. Since interpolation errors will be passed on to the next image, *k+2*, the image will be blurred. The synthesis of image *k+2* would require another interpolation of the already interpolated values from image *k+1*. This can only be avoided if no interpolated values are used for the image synthesis. With subpixel-accurate displacement vectors, image synthesis based on an image memory is, therefore, not a good choice.

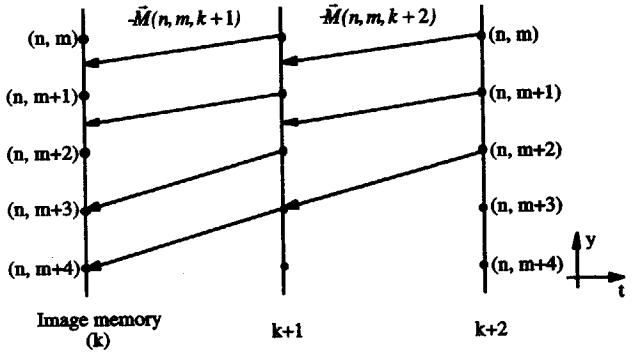


Figure 4.14: Image synthesis based on an image memory [Höt92]

Another problem - especially with stereoscopic coders - occurs when new regions appear. When new regions are added to the image memory, they might overwrite other information required at a later stage or for the second image of the stereo pair. This can happen, for instance, if a region is covered by another region. In an image memory, information about the covered part of the region would be deleted from the memory.

4.4.2 Image Synthesis based on a Region Memory

When image synthesis based on a region memory is implemented, the colour values to synthesise image $k+1$ are not taken from the previous image k, but from a region memory.

This region memory contains all the information of all the regions known up to that point in time - as is also the case with a texture map in graphics processors - but at a specific instant in time only a portion of this knowledge is used. As all the information for the synthesis of all images will now be taken from one common region memory, continuous interpolation of the colour information per region - even when using subpixel-accurate displacement vectors - can be avoided. In this way, blurring of the images can be suppressed. The principle of image synthesis based on a region memory is outlined in Figure 4.15. This one-dimensional situation with one region memory is again based on the example in Figure 4.14. First of all, the region memory has to be initialised. This is done by copying the contents of the first image which in this example is assumed to be image k to the region memory. To synthesise image $k+1$ and image $k+2$, the colour information of the region memory will be used - taking into account the displacement vector fields - without changing the contents of the region memory.

The motion vector field $\vec{M}(n,m,k+1)$ can be used directly to synthesise image $k+1$ from image k, as the region memory has been initialised with the data of image k. The methods of image synthesis, based either on an image memory or a region memory, are, therefore, the same for the first synthesis step.

To synthesise the next image $k+2$, the displacement must be calculated relative to the temporal position of the data in the region memory to obtain the relationship of image $k+2$ and the region memory. This is done by adding the motion vector fields. The new vector fields are indicated by the index *RM (Region Memory)*. In Figure 4.15, the new motion vector field $\vec{M}_{RM}(n,m,k+2)$ was calculated by adding $\vec{M}(n,m,k+1)$ and $\vec{M}(n,m,k+2)$.

When using displacement vectors with pixel accuracy, this addition can be performed without errors. By using the region memory instead of the image memory, the interpolated values at $(n, m, k+1)$ and $(n, m+1, k+1)$ will not be used for the synthesis of image $k+2$. An accumulation of synthesis errors due to the interpolation of the colour information is, therefore, avoided.

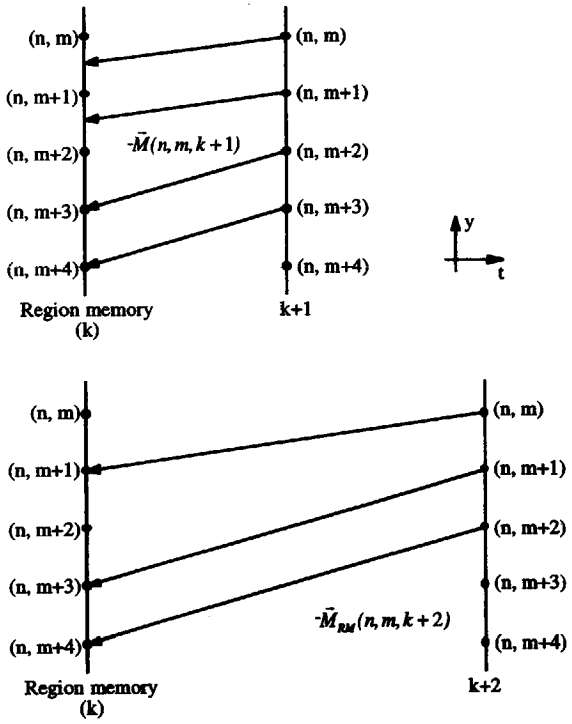
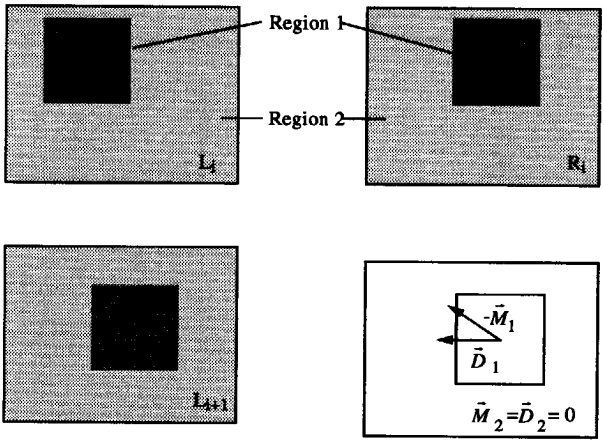


Figure 4.15: Image synthesis based on a region memory [Höt92]

In contrast to an image memory, a new region can simply be added to the region memory without changing or deleting existing information. Also, if a region is changed as a result of merging with another new region, this is easy to handle and the parameters in the region memory will be changed accordingly. The problems with a stereoscopic coder, addressed in the previous Section in relation to an image memory, are, therefore, non-existent when a region memory is used.

4.4.3 Using Motion and Disparity in Image Synthesis

The principle of region-based synthesis will be illustrated with the following example. Figure 4.16 shows the three images L_i, L_{i+1} and R_i and the parameters C, $\vec{M}$, S and $\vec{D}$ of their regions, which are determined from image analysis and stored in the region memory.



Colour-, Shape-, Motion- and Disparity-parameters of the region memory

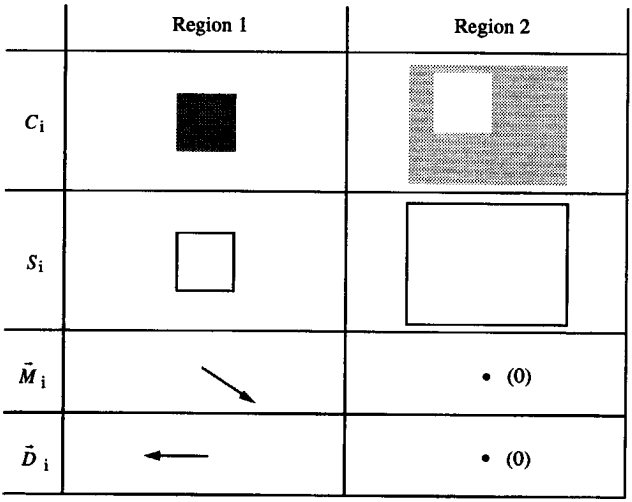


Figure 4.16: Depiction of the regions and their parameters in an example with two regions

Only two regions are present in this example, one - the square - is moving, the other is a stationary background. Image synthesis reconstructs the regions and shifts them according to the motion and disparity vectors. In this example, it is assumed that L_i has already been segmented into regions and the region parameters are stored in the region memory.

First of all, the left image is reconstructed by means of motion compensation. The resulting synthesised image L'_{i+1} is shown in Figure 4.17, where the white area is the uncovered background that was not visible in the previous image.

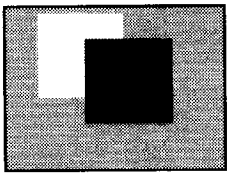
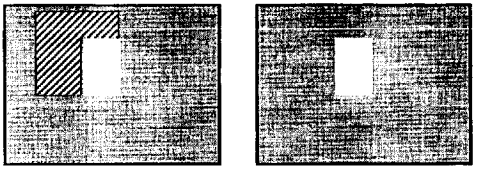


Figure 4.17: Resulting image after motion-compensation of the left image

This uncovered background will be treated as a new region by image analysis. The first step is to describe the shape of the occluded area, then extract the colour parameters from the original left image. At this stage, this area does not have any motion or disparity vectors assigned to it; these parameters will, therefore, be considered to be *unknown*. The new region will be transmitted and reconstructed at the same position as in the original image. The result is shown in Figure 4.18 (a) (for clarity, the square is not shown) where there are actually two regions now, the background of region 2 in image L_i and the newly transmitted, and now uncovered, background. At this time, the parameters of region 2 are still the same as shown in Figure 4.16. In the next analysis step, analysing the images L_{i+1}, L_{i+2} and R_{i+1}, it can be seen that the newly transmitted background and region 2 have the same motion and disparity, the two regions will then be merged. As the motion, disparity and shape parameters will not change, only the colour parameters will be updated at that time (Figure 4.18 (b)).



(a) (b)

Figure 4.18: Resulting image after transmission of the uncovered background (the square of Figure 4.17 is not shown)

With the aid of disparity information, the right image R_i will be reconstructed next. As region 2 is still described in the region memory as shown in Figure 4.18 (a) - merging region 2 and the uncovered background will take place in the next analysis step - the result of the stereo compensation is as shown in Figure 4.19 with the white area as an occlusion.

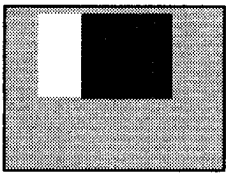


Figure 4.19: Resulting image after stereo compensation of the right image

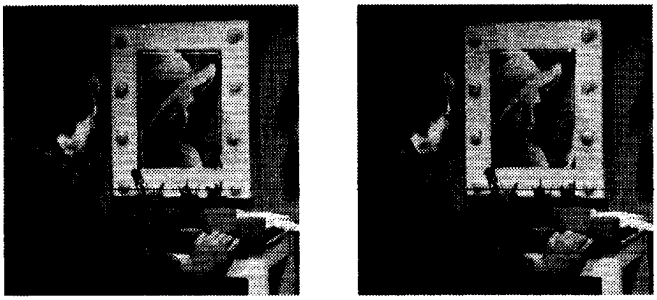
As in the motion-compensated case previously, the occluded area will be treated as a new region by image analysis. In the next step, this occlusion is already known to the receiver and sot does not have to be transmitted again.

4.5 Experimental Evaluation of Image Analysis and Synthesis

To assess the quality of the segmentation process, the percentage of correctly segmented pixels was investigated for the "Mirror" sequences. In these sequences, the position and size of the regions are known which means that the result of the segmentation process can be compared with a perfect segmentation. The experiment has been performed on image pairs without taking temporal correlation into account. This forces a complete segmentation in all of the image pairs. The segmentation results for the 5th and the 30th left image of "Mirror" with 30dB noise are shown in Figure 4.20. With these images, two regions have been segmented, Lena and the Clown, to be distinguished by their disparity. The percentage of correctly segmented pixels *(PCS)* has been evaluated as the number of pixels segmented as part of Lena and really belonging to that region, divided by the number of pixels for the actual region. In the first image, where there is no disparity or motion, the whole image is segmented as a single region. Therefore, as there is no region for Lena, evaluation of the correctly segmented pixels does not make sense. Starting with the second image, disparity information is available to perform a proper segmentation. About 93% of the pixels are correctly assigned to Lena independently of the disparity values and the amount of occlusion.

As could be seen in Figure 3.23 inaccuracies in the disparity estimation on "Mirror" occurred at the boundary of Lena towards the Clown. The 7% of incorrectly segmented pixels therefore

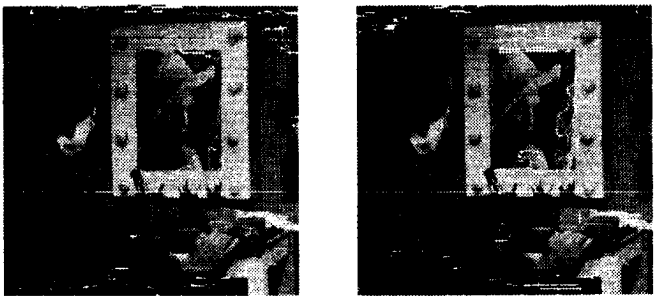
can only occur at this boundary. Although the shape description enlarges the region, this leads to contours on the inside of Lena as can be seen in Figure 4.20.



(a) (b)

Figure 4.20: Segmentation result for "Mirror" with 30 dB noise (left image number 5 (a) and left image number 30 (b))

This experiment was also performed using the "Mirror" sequence with 20dB noise. The noise decreased the quality of the disparity estimation, so making the segmentation more difficult. As can be seen in Figure 4.21, more than two regions have been identified by the segmentation process. When the *PCS* for Lena with these noisy images was calculated, all of the regions belonging to Lena were taken into account. In this way, about 91% of the pixels are correctly assigned. Bearing in mind the results of the disparity estimation (Figures 3.36 and 3.37) with a difference of about 10% of correct disparity vectors in these sequences, the segmentation process has proved to be robust as far as noise is concerned.



(a) (b)

Figure 4.21: Segmentation result for "Mirror" with 20 dB noise (left image number 5 (a) and left image number 30 (b))

Figure 4.22 finally shows the *PCS* for all the images of the sequences.

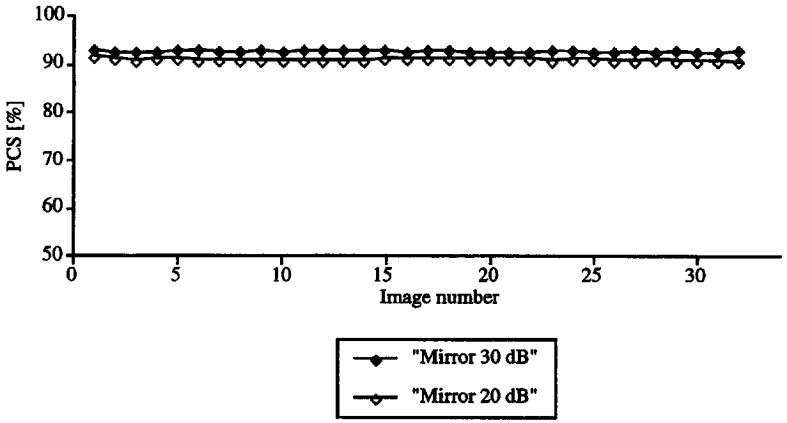


Figure 4.22: The percentage of correctly segmented pixels for Lena in the "Mirror" sequences

To assess the quality of the entire image analysis and image synthesis chain, the segmented regions of the first analysis step were used to synthesise the next left image with the aid of motion information and to synthesise the corresponding right image with the aid of disparity information. As described in Section 4.4.3, the regions have now been updated in this experiment. However, this was only possible for the left image sequence, where a segmentation of the uncovered background was possible with the aid of motion and disparity information when available in the later images. For occluded areas in the right image sequence this is not possible, as disparity can not be estimated with occlusions. In this experiment, the PSNR for the right channel was therefore calculated without taking occlusions into account. Consequently, large occluded areas gave a low PSNR, as no information was copied to such areas, but zero was assigned to them as their value. The synthesis error was evaluated by subtracting the synthesised image from the original image. By adding this error to the synthesised image, the image analysis and synthesis started with the original images but still used the regions and their information. In this way, the entire image analysis and image synthesis chain could be evaluated without taking the coding aspects into account.

Figure 4.23 shows the PSNR values (equation (3.24)) of the synthesised images without the synthesis error for the "Aqua" sequence. As only a small amount of motion occurs in "Aqua", the left synthesised image is of high quality. The exact PSNR values again depend on the amount of uncovered background, but also on whether new objects could be defined to

increase the quality. In the case of regions combining more than one disparity value, the quality of the right synthesised images is, of course, lower than for the left ones. As can be seen in Figure 4.23, the PSNR values for the right images are almost constant for the entire sequence. This is due to the fact that there is not much difference in the number of occluded areas from one image to another.

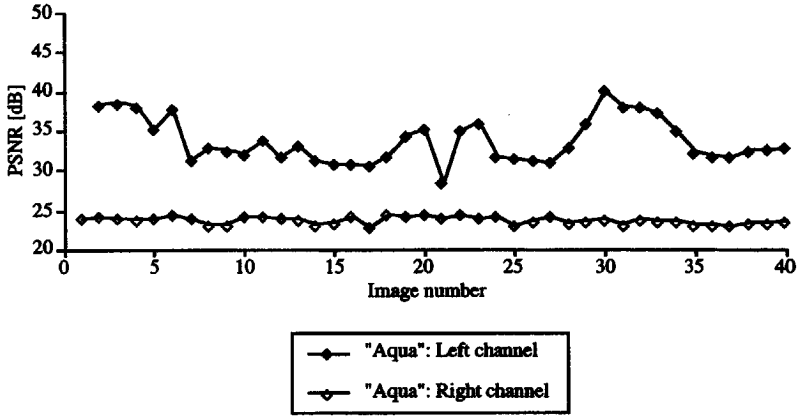


Figure 4.23: PSNR values for the left and right channel of "Aqua" after image analysis and synthesis

4.6 Conclusion

In this Chapter, the image analysis and synthesis tools to be used in the region-based stereoscopic coder have been presented. After the $2\frac{1}{2}-D$ regions have been segmented and described, several steps are implemented to reduce the number of regions. The regions' parameters - shape, colour, motion and disparity - are then stored in a region memory for the synthesis process. The region memory permits the creation of a database of regions present in the image sequence, so decreasing the bit rate required for transmission if, say, a previously visible region becomes covered and visible again. The regions will only be changed if new regions - "black holes" in the synthesised image - can be merged with them. For that reason, temporal consistency of the regions can be guaranteed. Image synthesis is a simple process because the regions will only be shifted according to their motion and disparity; any synthesis errors that occur are not yet dealt with in the synthesis process itself. The coding of the parameters and the treatment of synthesis errors will be described in the next Chapter.

5. Region-Based Stereoscopic Image Sequence Coder

The image analysis process splits the image into a set of regions and extracts the parameters colour C_i, shape S_i, motion $\vec{M}_i$ and disparity $\vec{D}_i$ for each region $\Re_i$. Image synthesis then reconstructs an image using these parameters which are stored in a region-memory. If image analysis fails to describe the scene completely, say, because of imperfections in the source model that is being used, there will be a synthesis error in the reconstructed image. In this Chapter, the concept of a region-based image sequence stereoscopic coder will be presented. Disparity estimation, image analysis and synthesis are used, but the transmission of the synthesis error is also taken into account. Also, the efficient coding of the parameters will be discussed, as well as a possible rate control for a region-based stereoscopic coder. Finally, the results of the region-based stereoscopic coder that has been described will be statistically and subjectively evaluated.

5.1 Concept of the Region-Based Stereoscopic Coder

Figure 5.1 shows a block diagram of the sender site of the region-based stereoscopic coder in detail. Based on Figure 2.10, an extension has been added to form a stereoscopic coder. In the *Image Analysis* module, the regions are segmented using the motion- and disparity vector fields estimated in the *Motion Estimation* and *Disparity Estimation* modules.

Each of the resulting regions is assigned its parameters colour C_i, motion vector $\vec{M}_i$, shape S_i and disparity vector $\vec{D}_i$. The colour parameter C_i contains the luminance and chrominance values of the region surface, the motion vector $\vec{M}_i$ describes the horizontal and vertical motion of the region. The shape parameter S_i contains a description of the position of the region in the camera-plane and its boundary. Finally, the horizontal component D_h of the disparity vector $\vec{D}_i$ is the displacement of the region from the left to the right image in the camera-plane. All these parameters have to be coded efficiently. This is done in the *Parameter Coding* module. After the parameters have been decoded in the *Parameter Decoding* module, these parameter sets are named C', $\vec{M}'$, S' and $\vec{D}'$. The shape, motion and disparity information undergo lossless coding in accordance with the concept described in this Chapter and so $S' = S$, $\vec{M}' = \vec{M}$ and $\vec{D}' = \vec{D}$. *Parameter Decoding* would, therefore, not be necessary for these three parameter sets; the original parameters could also be used for

reconstruction. However, the large amount of colour data must be reduced for the colour parameter set C which will undergo lossy coding. In order to give a general block diagram, independent of the coding schemes that have been used, Figure 5.1 shows *Parameter Decoding* for all the parameter sets.

Unlike block-based coders, which require an image memory to store the last image of the sequence to be coded and transmitted, a region-based coder needs a *Region Parameter Memory* for the transmitted and decoded parameter sets C', $\vec{M}'$, S' and $\vec{D}'$ for all the regions.

The memories of the sender and the receiver contain the same parameters, so enabling the sender as well as the receiver to synthesise the image using the same parameters. The synthesised image is displayed at the receiver site, but is also used to analyse the next image.

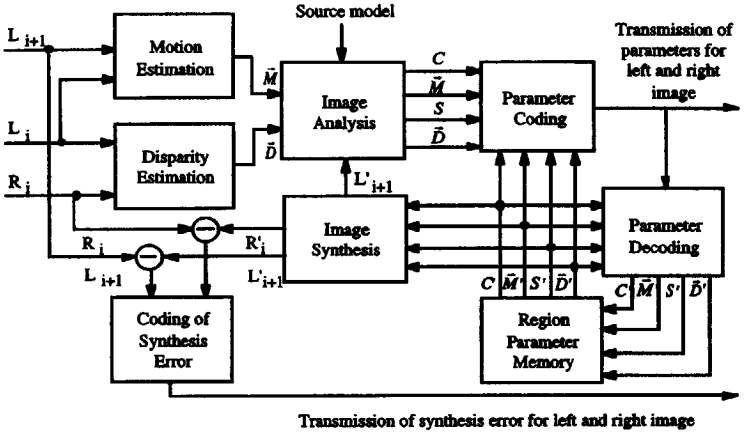


Figure 5.1: Detailed block diagram showing the sender site of the region-based stereoscopic image sequence coder

In the *Image Synthesis* module, the synthesised image is generated with the parameter sets C', $\vec{M}'$, S' and $\vec{D}'$. The synthesised left image is also sent to the *Image Analysis* module which will perform a region-based motion estimation on the synthesised image and the next original image using the region parameters C_i', S_i' and $\vec{M}_i'$ and a region-based disparity estimation on the synthesised image and the original right image using the region parameters C_i', S_i' and $\vec{D}_i'$ in the next coding step. In this way, the vector information is always based on the images which are also present at the receiver site, so avoiding error accumulation due to inaccurate vector estimates transmitted from the coder.

As image synthesis fails if the source model is not correct or if image analysis introduces inaccuracies, a way of correcting such failures and inaccuracies has to be included. This is done in the *Coding of Synthesis Error* module.

5.2 Coding of the Region Parameters

In the *Parameter Coding* block in Figure 5.1, all parameters are encoded as efficiently as possible to achieve a coding gain. Two different *Parameter Coding* cases have to be distinguished: either regions will be transmitted for the first time, or the parameters in the memory will be updated when a region already exists but some of the parameters have changed. Depending on the case, either temporal or spatial correlation within each parameter stream can be exploited. To further increase the efficiency of parameter coding, interrelations between different parameter streams, say shape and motion parameters, can also be exploited. Figure 5.2 shows the principal methods of parameter coding for existing-region updates which will be discussed in this Section.

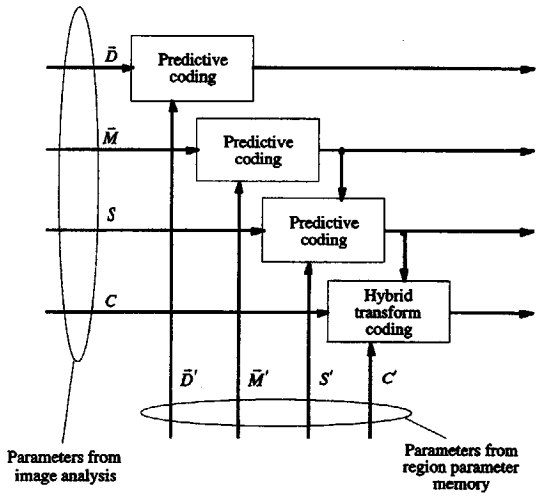


Figure 5.2: Parameter coding for updating regions that already exist in the region-based stereoscopic image sequence coder

Figure 5.2 shows that the parameter sets $\vec{M}$ and $\vec{D}$ are encoded independently of all the other parameter sets, where S depends on the motion vector of the region and C depends on its shape. Since the disparity information is not used to encode any other parameter, the system

can also be used for monoscopic sequences. This is also obvious from the concept in Section 5.1, where the left image is synthesised first, and the synthesis of the right image is based on the left image. This also includes the colour parameter C which contains only the colour information of the left images and is used synthesise the right images as well.

5.2.1 Coding of Motion and Disparity Parameters

When $2\frac{1}{2}-D$ regions are used as a source model as described in Chapter 4, the motion and disparity vectors for each region are stored in the region-parameter memory. Lossy coding of motion and disparity information would lead to large synthesis errors - or even to an incorrect three-dimensional impression when disparity vectors are involved. Therefore, the vectors have to undergo lossless encoding using redundancy reduction only. As the vectors are associated with regions and not with blocks, there is no spatially regular vector field. Methods such as lossless DPCM [ZTS94] can, therefore, not be applied. When new objects are transmitted, there is no way of using temporal prediction of the vectors either.

One way of encoding the motion and disparity vectors in a region-based stereoscopic coder is spatial prediction of the vector of region $\Re_i$ from the vector of region $\Re_{i-1}$. This means that only the first vector in an image will be transmitted directly. Subsequently, only the differences $\vec{M}_i$ - $\vec{M}_{i-1}$ and $\vec{D}_i$ - $\vec{D}_{i-1}$ will be transmitted. A second approach is not to predict the vectors at all, but to use only Huffman coding. Figure 5.3 shows a histogram of the disparity values $\vec{D}_i$ (Figure 5.3 (a)) and the differences $\vec{D}_i$ - $\vec{D}_{i-1}$ (Figure 5.3 (b)) from the "Aqua" sequence as an example.

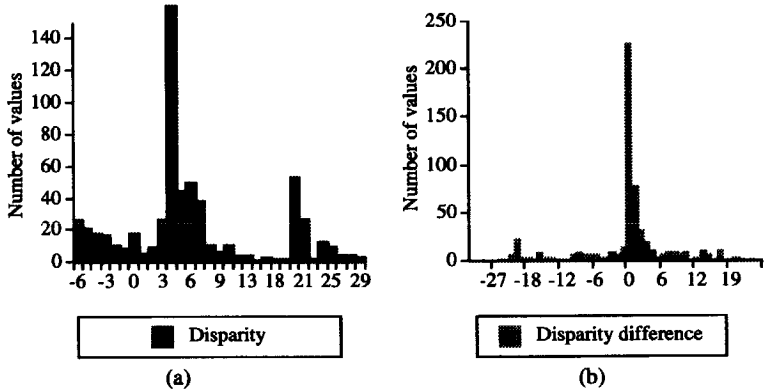


Figure 5.3: Histograms of the disparity values for newly identified regions in the "Aqua" sequence: (a) $\vec{D}_i$ and (b) $\vec{D}_i$ - $\vec{D}_{i-1}$

In these histograms, all disparity values for newly identified regions in the entire sequence are taken into account. These are all the regions in the first image pair and all uncovered and occluded regions in the subsequent image pairs described in image analysis. In the first image pair, the regions are ordered according to the size of their disparity vector. Therefore, a high number of neighbouring regions have a disparity difference of zero (see Figure 5.3 (b)). With subsequent image pairs, the regions will be numbered according to the occurrence of the new region. Therefore, the correlation between vectors of neighbouring regions becomes less, which can be seen in Figure 5.3 (b) as there are a number of large disparity differences.

The basis for deciding whether to use direct or differential encoding is an evaluation of the entropy H:

$$H = -\sum_{d=d_{min}}^{d_{max}} p(d) \cdot \log_2[p(d)] \tag{5.1}$$

where $p(d)$ is the probability of the symbol d in the set of data, calculated from the histograms in Figure 5.3.

The entropy gives the theoretical minimum number of bits required to code a vector component using redundancy coding. Table 5.1 gives the calculated entropy values of the different vectors for newly identified regions in the "Aqua" sequence, based on the histogram data for the disparity vectors as shown in Figure 5.3.

	Horizontal Motion	Vertical Motion	Disparity
Spatial prediction	1.6	0.0	4.1
No prediction	1.3	0.0	4.0

Table 5.1: Entropy of vectors for newly identified regions in the "Aqua" sequence

As there is no vertical motion at all in "Aqua" and the horizontal motion is basically constant, the entropy of motion vectors is rather low compared with that of disparity vectors. In the case of disparity vectors in particular, differences can be quite high from one region to the next so giving entropy values which are somewhat higher than those for motion components. Based on the values in Table 5.1, vectors for newly identified and transmitted regions will only be entropy encoded and not predicted. In the system described in this thesis, Huffman coding, a common entropy coding method, is used.

When updating already existing regions the behaviour is different. The vector components to be transmitted are relative to the position of the region in the region memory. Figure 5.4

shows the motion components and the disparity values for a typical region as they are stored in the region memory at each point in time of the sequence. Without prediction, all these values would have to be transmitted as shown in Figure 5.4. For the horizontal motion component in particular, this would result in high entropy as there are 40 different values to transmit. This means that the required bit rate will also be quite high.

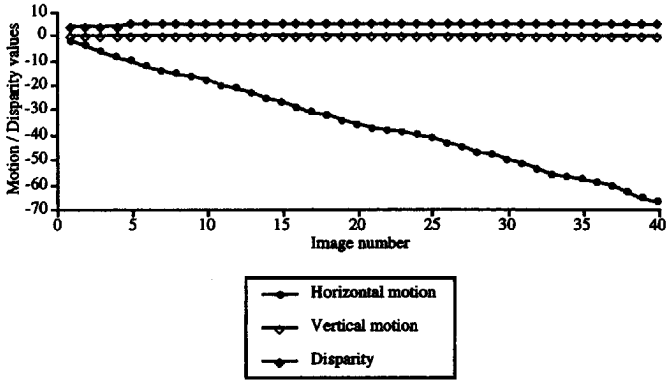


Figure 5.4: Motion vector components and disparity stored in the region memory

When exploiting the temporal prediction of the vector components, the entropy, and so the bit rate, can obviously be reduced. Figure 5.5 shows the differences after temporal prediction of the current vector components from the content of the region memory which is assumed to be set to zero initially.

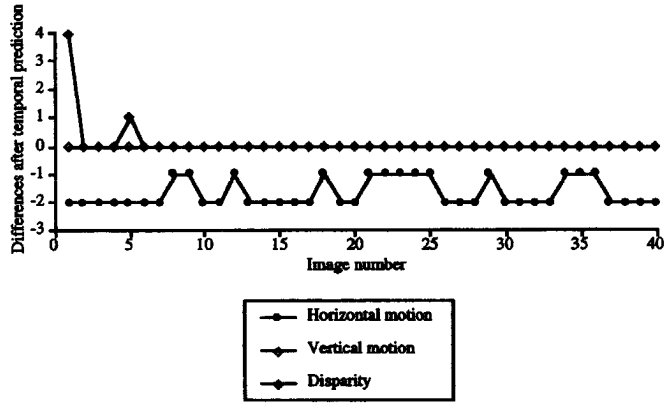


Figure 5.5: Differences after temporal prediction of the current vector component from the region memory

In this case, the vertical motion component M_v is zero for the region in all the images. Except when the region is initially defined, no information needs to be transmitted for the region in the sequence. The situation for disparity is almost the same - except for the first and fifth image no information needs to be transmitted.

Obviously, the horizontal motion component M_h is more difficult to handle. As in "Aqua" all the regions move horizontally, information has to be transmitted for every image. What will be transmitted is the difference between the value in the region memory and the new value to be stored as shown in Figure 5.5.

5.2.2 Coding of Shape Parameters

After image analysis is performed, all regions are described in terms of their shape parameters, using polygon approximation. To synthesise successive images, the shape is shifted according to the motion of the region. To synthesise the corresponding right images, the regions are shifted according to their disparity.

According to the source model, only rigid regions are allowed. This means that it is not necessary to update the region description in this system. Therefore, the parameters for coding a contour need to be transmitted just once.

Whenever a shape has to be transmitted, it is not necessary to transmit all the contour points, but only the vertices of the polygon. A straightforward solution is the transmission of the absolute spatial positions of all the vertices. The histogram of what co-ordinates would have to be transmitted in the "Aqua" sequence is shown in Figure 5.6. In this Figure, no distinction is made between the x and y co-ordinates. As uncovered background (which will be handled as a new region on its own) due to motion in "Aqua" mainly occurs at the right border of the image, a large number of vertices with large co-ordinates have to be transmitted.

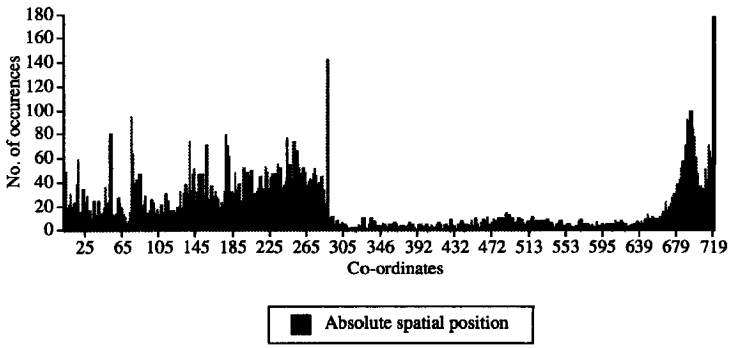


Figure 5.6: Histogram of the absolute spatial co-ordinates of the shape vertices in "Aqua"

A second solution is to transmit only one absolute spatial position of a starting vertex and the relative spatial position of the other vertices relative to each of the previous vertices of the polygon. The corresponding histogram is shown in Figure 5.7, where the differences of the x co-ordinates and the differences of the y co-ordinates of the vertices are plotted.

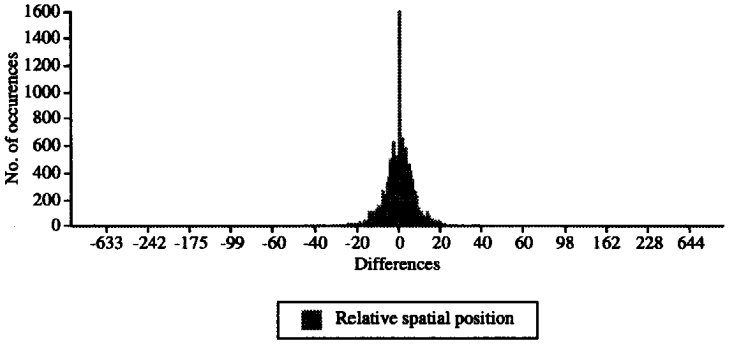


Figure 5.7: Histogram of the relative spatial co-ordinates of the shape vertices in "Aqua"

The entropy H (as given by equation (5.1)) has been calculated for both methods to decide which method needs less bits to transmit the region shape. The data shown in Figure 5.6 give an entropy of 8.7 bits per vertex co-ordinate, whereas the data from Figure 5.7 an entropy of only 5.3 bits per vertex co-ordinate. Therefore, the decision in this system was to transmit the vertices of the shape using relative addressing and applying Huffman coding to the differences to reduce the redundancy. This principle is also used in other systems such as the MPEG 4 video verification model [MPEG96, MPEG97]. Figure 5.8 illustrates the principle of relative addressing with the shape already shown in Figure 4.12.

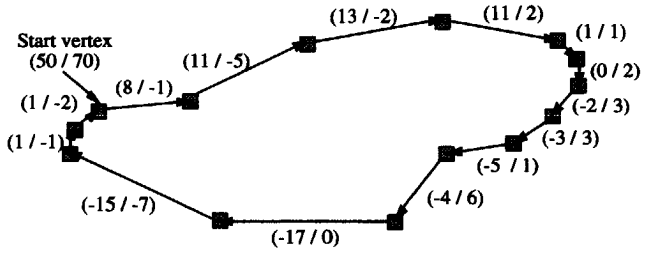


Figure 5.8: Relative addressing of the vertices of the shape from Figure 4.12

With the tolerances for the shape description as described in Chapter 4 and an average number of 300 contour points per region in the "Aqua" sequence, this principle gives an

average number of 1.6 bits per contour point. This value does not differ from results from similar approaches described in the literature [EK85, Ste93].

5.2.3 Coding of Colour Parameters

To code luminance and chrominance information, referred to as colour parameters, of arbitrarily shaped regions, methods which are not limited to fixed blocks of pixels are required. In the system which is being described, colour parameters are transmitted together with other parameters when the region is written to the region memory. This happens whenever a new region is defined, say for the first image pair, for uncovered background in the subsequent left images or for occlusions in the subsequent right images, but also for newly combined regions. When two or more regions are merged, the new contour for the new region is described. Due to the tolerances in the shape description, this could lead to a situation where not all the colour parameters for all the pixels inside the new region are described. To avoid such problems, the colour parameters of the new region are transmitted and added to the region memory.

Several non-block colour coding methods are known from the literature [Phi96]. One approach is to approximate the luminance and chrominance values with polynomial functions [Koc83, Leo87]. When a lot of luminance and chrominance changes have to be coded within the region, high-degree polynomial functions which are difficult to define have to be used. As segmentation is only based on motion and disparity, but not on colour, this occurs very frequently. A second solution is region-oriented transformation coding [GEM89]. Finding the appropriate orthogonal functions is a computational very expensive procedure using orthogonalisation schemes. Extrapolation [KA93, Kau95] aims at a shape-independent description using a circumscribing rectangle. To do this a computational expensive regularisation is necessary. For these reasons, shape-adaptive DCT [SM95] was used in this thesis. On one hand, correlations are not completely exploited, but on the other hand a pre-defined orthogonal set of DCT basis functions can be used, which makes the algorithm easy and fast [SBM95, Sik96]. It is also the choice of the MPEG 4 video group [MPEG97].

Figure 5.9 shows the basic principle of shape adaptive DCT (SA-DCT). Figure 5.9 (a) shows an example of an image block segmented into two regions. To perform a vertical SA-DCT transformation of the dark region, the length (vector size V, $0<V<9$) of each column j ($0<j<9$) of the region is calculated. The columns are then shifted and aligned with the upper border of the 8x8 reference block (Figure 5.9 (b)). Depending on the vector size V of each particular column of the region, a DCT transform matrix $\underline{DCT}_V$ (given by equation 5.2) containing a set of V $\underline{DCT}_V$ basis vectors is selected.

$$\underline{DCT}_V(k,l) = c_0 \cdot \cos\left[k\left(l+\frac{1}{2}\right)\cdot\frac{\pi}{V}\right] \tag{5.2}$$

where $0 \le k \le V\text{-}1$ and $0 \le l \le V\text{-}1$

$c_0 = \sqrt{\frac{1}{2}}$ if k = 0

$c_0 = 1$ otherwise

k denotes the k^{th} DCT_V basis vector

The V vertical DCT-coefficients $\vec{c}_j$ for each set of region column data $\vec{x}_j$ are then calculated by setting $k=u$ according to [MPEG97, Kau97]:

$$\vec{c}_j = \frac{4}{V}\cdot\underline{DCT}_V\cdot\vec{x}_j \tag{5.3}$$

For example, in Figure 5.9 (b), the rightmost column is transformed using $\underline{DCT}_3$ basis vectors. The coefficients are shown in Figure 5.9 (c). To perform the horizontal DCT transformation, the length of each row is calculated and the rows are shifted to the left border of the reference block (Figure 5.9 (d)). A horizontal DCT with $k=v$ adapted to the size of each row is then calculated using equations (5.2) and (5.3). Figure 5.9 (e) shows the final location of the resulting DCT coefficients.

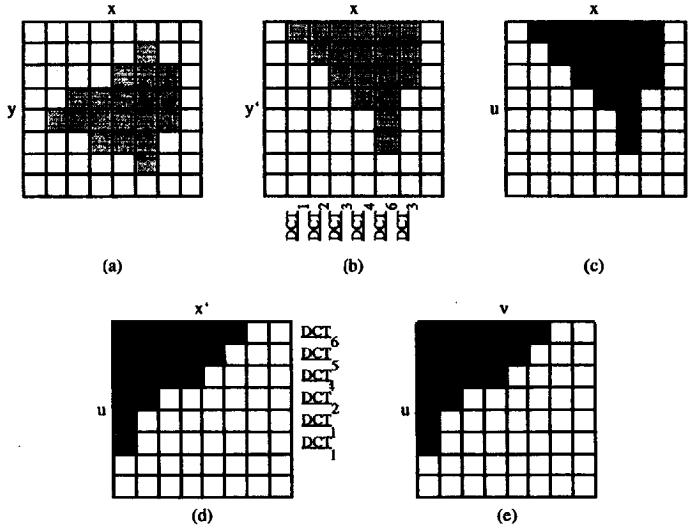


Figure 5.9: Successive steps involved in performing shape-adaptive DCT [SM95]

The final number of DCT coefficients is equal to the number of pixels in the region. To obtain a coding gain, the coefficients are quantised with a fixed quantiser and an identical quantiser step-size for all the coefficients. The coefficients are scanned using zigzag scanning as shown in Figure 5.10.

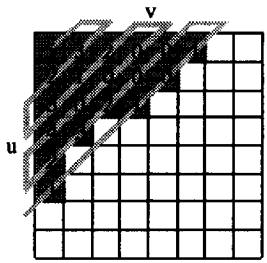


Figure 5.10: Zigzag scanning of DCT coefficients

Next, the coefficients are run-length encoded. For this, a run-length, indicating the number of steps in the order of the zigzag scan to the next non-zero value, will precede the value of the coefficient. Pixels outside the region to be encoded are not taken into account. They are therefore skipped when the run-length is calculated. In the final step, the run-length code is Huffman-encoded using two different Huffman tables as in the H.261 standard [GSF95], one for the coefficient values and another for the run-length values. The DCT coefficients used as an illustration in Figure 5.10 would therefore give the results shown in Figure 5.11, where the Huffman coding is not yet included:

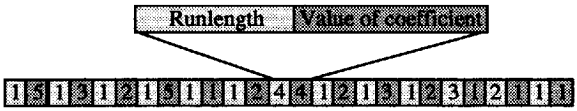


Figure 5.11: Example illustrating run-length coding (coefficients taken from Figure 5.10)

A single, combined Huffman table for the pair {run-length, value of coefficient} would also be possible. This would actually improve the performance of the colour parameter coding. On the other hand, the creation of a single table of this kind is more complicated and the table would also be larger. The decision was, therefore, to use two different Huffman tables as described above.

The decoding of the transmitted values at the receiver [MPEG97] is possible since the shape of the region as shown in Figure 5.9 (a) is known. The receiver can easily identify the position of the coefficients by creating the shape of Figure 5.9 (a).

The reverse operation

$$\vec{x}_j = \underline{DCT}_V^T \cdot \vec{c}_j \tag{5.4}$$

in both the horizontal and the vertical direction and reverse shifting of coefficients will reconstruct the region (from Figure 5.9 (e) to Figure 5.9 (a)).

SA-DCT is used to code the colour parameters of a region by splitting the region into fixed 8x8 blocks. All the blocks are coded according to the scheme described above. If a block is on the border of the region, SA-DCT is implemented as described above using the appropriate basis vectors. If a block is completely inside the region, SA-DCT is implemented using $\underline{DCT}_8$ basis vectors only, so acting like a "normal" 8x8 DCT.

5.3 Coding of the Synthesis Error

The system described in Chapter 4 will not be able to synthesise a perfect image in the sense of an image that is identical to the original. Due to quantisation of the colour parameters, there will be differences inside the regions. The tolerance of the shape description will result in an inaccurate description of the region boundaries. Finally, the source model that has been adopted will limit the success of image synthesis if the model is not accurate. All these inaccuracies will produce errors in the synthesised images (referred to as synthesis error) which should be corrected using as few bits as possible or even no bits at all in an optimal case.

As a human viewer concentrates on a natural displacement, but not on the exact positioning of a region in the image. Small positioning- and shape-errors are regarded as being irrelevant to the human viewer [Höt92]. In a region-based coder as described here, the goal is high-quality image synthesis with respect to the subjective visual quality for a human viewer to minimise the number of bits required to transmit a synthesis error. If synthesis were perfect in this sense, it would not be necessary to transmit any synthesis error at all.

Figure 5.12 shows synthesised, right image 5 of the "Aqua" sequence. In the image that is shown, the occluded areas have already been transmitted as described in Section 4.4.3.

Looking at this single image, no disturbing artefacts can be seen. However, when the differences between this synthesised image and the original image (shown in Figure 5.13) are examined, with an offset of 128 added to the difference values, it is obvious that the synthesis is very far from being perfect in the sense of pixel-accurate prediction.

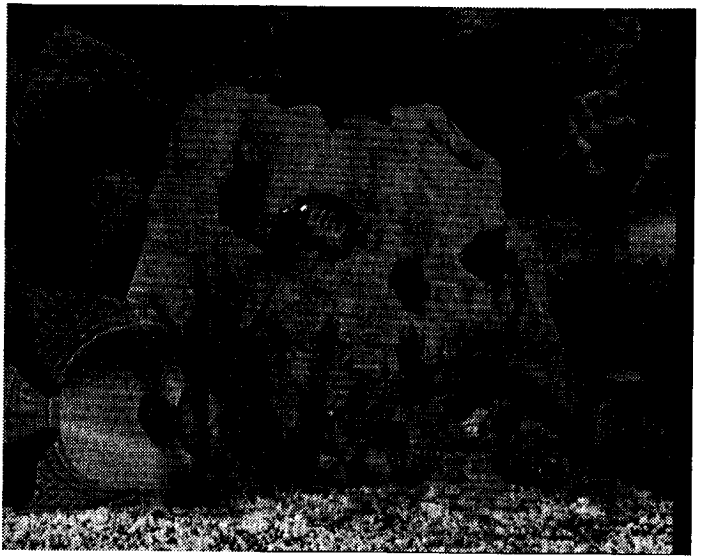


Figure 5.12: Synthesised right image 5 of "Aqua"

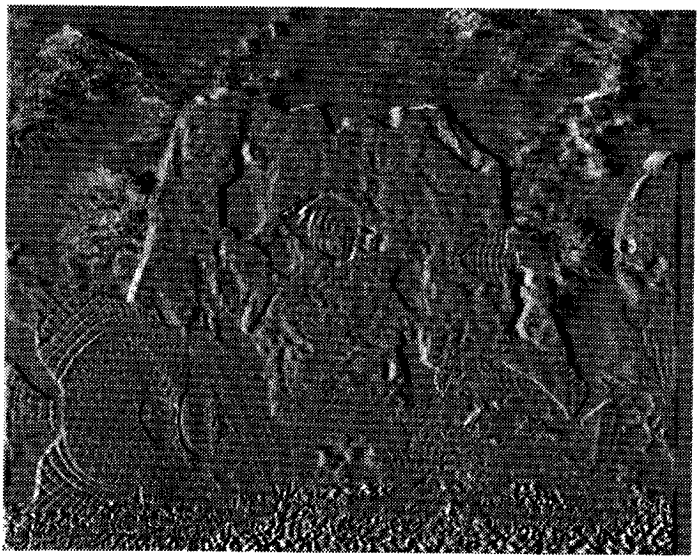


Figure 5.13: Differences between the synthesised right image 5 of "Aqua" and the original image (offset 128)

Obviously the PSNR-value of a synthesised image of this kind will be relatively small, especially in comparison with block-based techniques. This is due to misalignments of the regions which do not influence the visual quality of the image. An enlargement of a critical Section, the big fish in the lower left corner of Figure 5.12, shows the reason for the small PSNR-values. This enlargement is shown in Figure 5.14.

(a) (b) (c)

Figure 5.14: Enlargement of a high-error area:
(a) original, (b) synthesised, (c) difference (offset 128)

Even with such an enlargement no disturbing errors can be seen. The only reason for the huge errors shown in Figure 5.14 (c) is a positioning error of the regions of about 2 pixels, coming from inaccuracies in the motion estimation and tolerances of the shape description. If the objective is to have good subjective visual quality, there is no need to transmit such errors for a single image.

When dealing with the region-based coding of image sequences, one of the most important topics concerning visual quality is the temporal behaviour of the regions. Since only rigid regions are allowed in this system, temporal consistency of the regions themselves can be guaranteed. However, consistent, smooth motion of regions cannot be guaranteed. This can be seen in Figure 5.5 where the horizontal motion alternates between values of -1 and -2 from one image to the next one. This is due to the use of only pixel-accurate vectors but also to estimation inaccuracies. In the example-region shown in Figures 5.4 and 5.5, which moves 66 pixels to the left in 40 images, the value needed for smooth horizontal motion in "Aqua" would be -1.65 pixels. In an image sequence such alternating values would result in an unnatural movement of the regions when motion vectors are concerned, and to inconsistent positioning of regions in the right images when disparity vectors are concerned. As long as temporal inconsistencies like this can occur, the transmission of a synthesis error is necessary to minimise these effects.

The problem with the synthesis error is knowing which parts have to be transmitted and which parts can be omitted. As discussed above, it is not very important to transmit luminance or chrominance errors inside a region, but, in terms of temporal consistency, a precise positioning of the regions becomes more important. As can be seen in Figure 5.13, the error is especially high at region boundaries. For this reason, the synthesis error has been transmitted whenever its absolute value was above a certain threshold. Figure 5.15 shows the error of Figure 5.13, again with a threshold of 20, which finally has to be transmitted. In the "Aqua" sequence, an average of 12% of the image has to be transmitted.

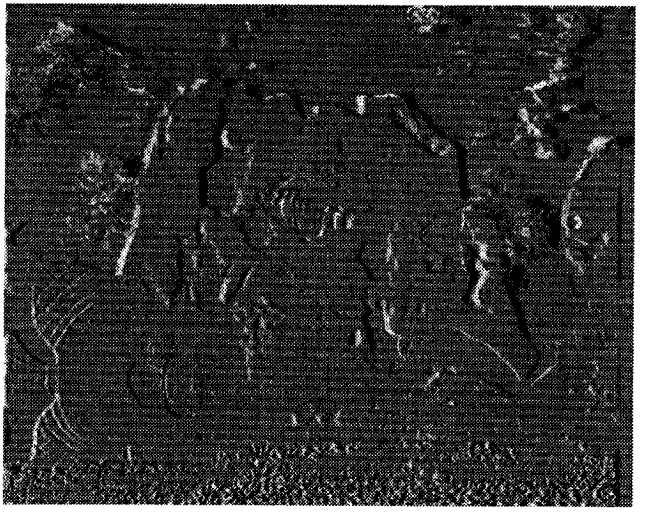


Figure 5.15: Differences between the synthesised right image 5 of "Aqua" and the original image (threshold of 20, offset 128)

To code this synthesis error, standard algorithms can be used. Obviously a block-based method such as MPEG2 error coding using DCT will require a very high data rate to code the error or result in block artefacts at a lower bit rate. This is due to the fact that the entire error image will be described as blocks and the data will be encoded, even if it is not necessary. When implementing a region-based coding scheme, it, therefore, cannot be the goal to introduce block-based coding schemes for the synthesis error.

A simple way of avoiding blocks and transmitting data only when really necessary is vector quantisation as described in [SIM94]. With the low expected data volume, a simple way of letting the decoder know where the information has to be added to is to transmit the co-ordinates of all the blocks in addition to the vector quantised information. The vector quantisation operates on a 2x2 luminance block and the corresponding chrominance values.

The aim is to find an appropriate code number from a set of pre-defined code-books. The code-books are defined in advance and are available to both the sender and the receiver. The code-books used in this thesis have been taken from [SIM94], therefore six different code-books with 32 to 1024 entries have been available. After the best code-book for the block to be encoded has been chosen, the number of the code-book that has been used and the appropriate code number for the data are transmitted. Since the code-books that are used have been trained on images that differ from those used here, this gives suboptimal coding behaviour. This means that fewer bits would be required for transmission if specially adapted code-books were available. However, experiments show the feasibility of the system and give an idea of the bit rate required to code the synthesis error. With the "Aqua" sequences, where all the regions move, 4.9 Mbit/s have been needed to encode the synthesis errors for the left and the right channel as described above. With "Tunnel", where only a small number of regions move compared to "Aqua", 3.9 Mbit/s were required.

5.4 Rate Control

An important advantage of region-based analysis-synthesis coding over block-based hybrid coding is that the block-based displacement estimation of a block-based coder is replaced by a pixel-wise displacement estimation and image analysis. Image analysis gives the opportunity to check the displacement description and to control the coding of the parameters on a region-basis. For each region segmented in image analysis, it can be decided which parameters have to be transmitted and which parameters can be skipped. However, as the whole concept is based on regions, the data-rate strongly depends on the number of segmented regions. With a high number of regions, the data-rate also will be high and with a small number of regions the image can be transmitted at a low data-rate. On this basis, the number of regions is the major key for controlling the total data-rate. In order to reduce the total bit rate significantly, it would be necessary to reduce the number of regions by merging them with neighbouring regions. This again results in an increase in the synthesis error, also increasing the number of bits necessary for its transmission.

As there are only a limited number of ways of fixing the bit rate, region-based coding schemes are inherently variable bit rate (*VBR*) coders. What can be restricted within certain ranges is the peak rate for transmission. A possible peak rate control is shown in Figure 5.16. When control of this kind is implemented, it is possible to adjust the peak bit rate to the requirements of the network. As one parameter influences the other, extensive tests within the rate control are necessary to find the optimum for a certain bit rate. In Figure 5.16 this is indicated by the loop rate control - parameter coding - quality evaluation. A change in the coding of one parameter will influence at least one other. This means that a rate control has to

check different settings and finally select the best one based on a quality evaluation of the result.

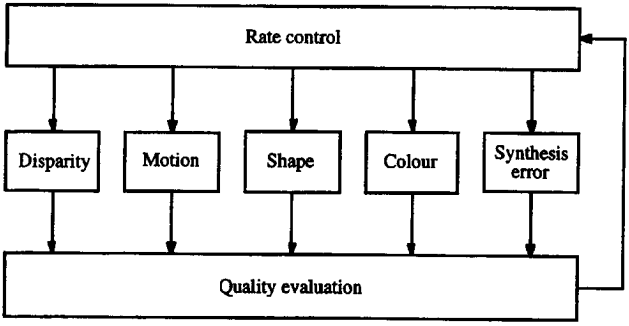


Figure 5.16: Possible peak rate control in a region-based stereoscopic coder

The goal of parameter control in a region-based coder must be to reduce the high amount of colour-coding without reducing the subjective quality. Two important characteristics of region-based coding support this idea:

- The quality of the displacement description at region-boundaries is improved by transmitting the region-shape. Each region now has its own displacement information, independent of its position in a block. Assuming error-free displacement vectors, the displacement compensation will, therefore, also be error-free at region-boundaries and no more colour parameters have to be transmitted in these areas.

- Image analysis might be able to detect small positioning- and shape-errors of the regions. As these errors do not influence the subjective quality, the coding and transmission of colour information can be suppressed in these areas.

A rate control has to distinguish four coding-modes: the update of regions, the coding and transmission of occluded areas and uncovered background as new regions, the coding and transmission of new regions for the first time and finally the coding and transmission of the synthesis error. Figure 5.17 shows the parameters to be transmitted depending on the coding-mode.

Parameter \ Mode	Mode 1 Update of already segmented regions	Mode 2 Uncovered background and occluded areas	Mode 3 New merged regions	Mode 4 Synthesis error
Disparity	X		X	
Motion	X		X	
Shape		X	X	
Colour (SA-DCT)		X	X	
Colour (VQ)				X

Figure 5.17: Parameters to be coded and transmitted in the various coding-modes

In mode 1 (*"Update of already segmented regions"*) it is necessary to transmit either the motion information, the disparity information or both if the region has changed and so no longer agrees with its description in the region memory. In mode 2 (*"Uncovered background and occluded areas"*) only the shape and the colour parameters have to be transmitted. In mode 3 (*"New merged regions"*) the complete region description has to be transmitted. Mode 4 (*"Synthesis error"*) only transmits the colour information using vector quantisation for a couple of pixels and this may not be necessary at all.

The first thing to be coded is always the displacement information (disparity and motion vectors) for the regions that have already been segmented. Then, the shape of all new regions is approximated and coded. Finally, the colour parameters of uncovered background, occluded areas and the synthesis error will be coded to allow predictive coding of the parameters as described in Section 5.2. In this way, the rate control can only influence the total bit rate by not transmitting new regions or by adjusting the colour parameters. As with "normal" DCT, the coefficients of the SA-DCT which has been used can also be quantised according to channel requirements. However, this decision will now be taken on the basis of regions and not on blocks, so avoiding the typical block-based errors.

Since the motion parameters, the disparity parameters and the shape parameters always have to be transmitted when a region-based coder is used, there is no way to achieve a constant bit rate (*CBR*) coder when working with regions. The bit rate can largely be adjusted by quantising the colour information. A second possibility, indicated above, is not to transmit new regions with their parameters but to handle them like synthesis errors. This means that coding the synthesis error would cover large areas of the images. Assuming that no regions are transmitted at all, this leads to a standard intra-frame coder as a fall-back solution.

5.5 Experimental Evaluation of the Region-Based Stereoscopic Coder

In order to evaluate the performance of the region-based stereoscopic coder, several investigations concerning the necessary bit rate and the resulting quality have been carried out. Firstly, a statistical analysis for the sequences "Aqua" and "Tunnel" is given, followed by an evaluation of the subjective quality compared with a stereoscopic signal, where both channels have been MPEG2-encoded separately.

5.5.1 Statistical Analysis

In this Section, the coding of the "Aqua" and the "Tunnel" sequences is statistically analysed to demonstrate the performance of the region-based coder. First of all, the rate used for the region parameters is investigated. These parameters are used to synthesise the images as described in Section 4.4.3. The peak signal to noise ratio (PSNR), calculated from equation (3.24), will be presented. Next, the synthesis error as described in Section 5.3 will be evaluated. The percentage of pixels to be transmitted as a synthesis error and the required bit rate will be investigated as well as the final image quality measured again in terms of the PSNR.

Figure 5.18 shows the bit rate used to code the parameters to be transmitted for both channels of the stereoscopic "Aqua" sequence. No rate control has been used in this experiment, so allowing the transmission of as many bits as is required for the shape, motion and disparity parameters and the use of a fixed quantisation for the colour parameters. Most of the bits are required to code the colour information of the regions using shape-adaptive DCT (SA-DCT). By far the most bits are required for the first image as all the regions have to be described for the first time. Whenever uncovered background or occluded areas are transmitted as new regions, additional bits will be used to describe these regions. In total 240 kbits are used in the "Aqua" sequence of 40 images with 720 pixels per line and 576 lines per image to transmit all the shape, motion and disparity parameters. An additional 700 kbits are required to transmit the colour information of the regions. In total, a bit rate of approximately 600 kbits per second is required for both channels of the stereoscopic sequence.

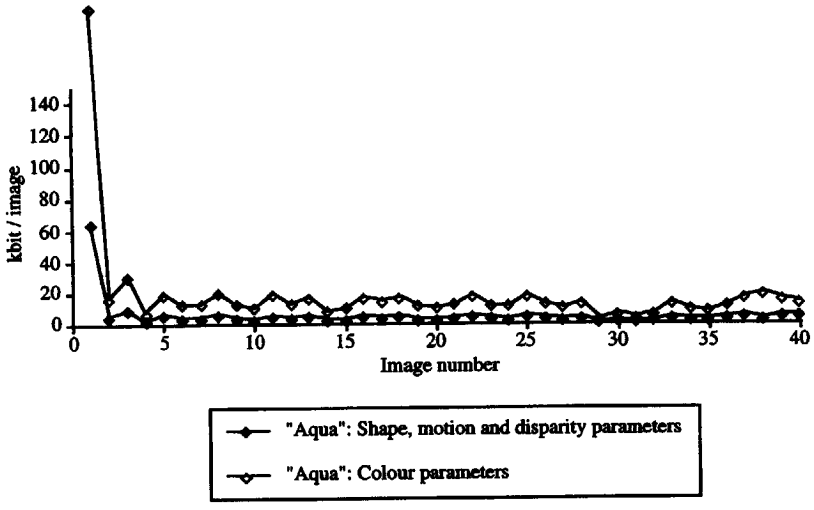


Figure 5.18: Bit rate used for the region description of the stereoscopic sequence "Aqua"

Using only these parameters to synthesise the images of the sequence results in the PSNR-values shown in Figure 5.19.

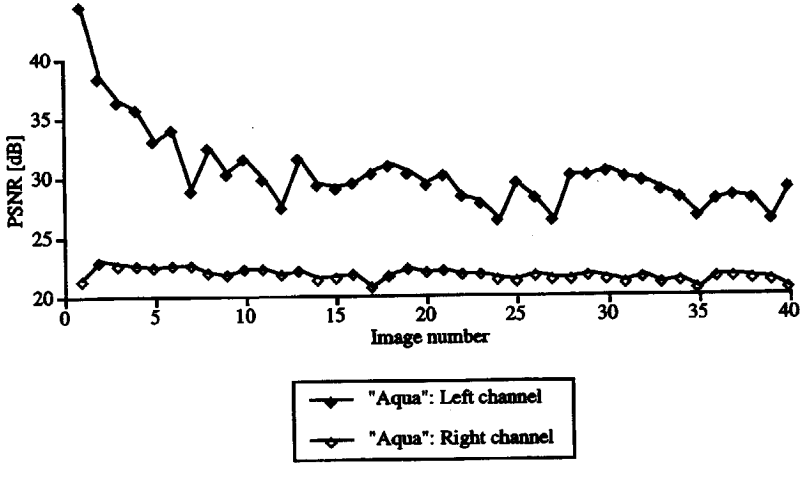


Figure 5.19: PSNR-values for synthesised images of "Aqua" using only the region parameters

The segmentation of the regions was based on the first image. The PSNR-value is, therefore, the highest for this image in the left channel. Starting with the second image, the regions have been shifted according to their motion vectors, so introducing motion estimation and compensation inaccuracies. The PSNR-values, therefore, decreased rapidly for the first couple of images until the maximum error was reached in image 7. The PSNR decreases further as the colour information of the regions is merely updated when regions are merged. Lighting changes are, therefore, not updated in "Aqua" where merging of regions is only performed for uncovered background. These areas of uncovered background occur only at the right border of the image, as there is a constant motion due to a camera pan from left to right. Because of the pixel accuracy of the motion estimator, this constant motion is not detected as constant as it is in reality (see Figure 5.5). Sometimes, this leads to rather large changes in the PSNR-values from one image to the next as can be seen in Figure 5.19.

For the right channel of "Aqua", the PSNR-values are lower than for the left channel. Since different disparity values are combined in the regions (the disparity compensation of regions for the right channel is, therefore, not as precise as the motion compensation used for the left channel) the quality of the right images is not as high as for the left images. Since these errors occur in all right images to more or less the same extent, the PSNR-values do not decrease in this case.

The percentage of pixels (based on the image size of 720x576 pixels) per image to be handled as a synthesis error, as described in Section 5.3, is shown in Figure 5.20.

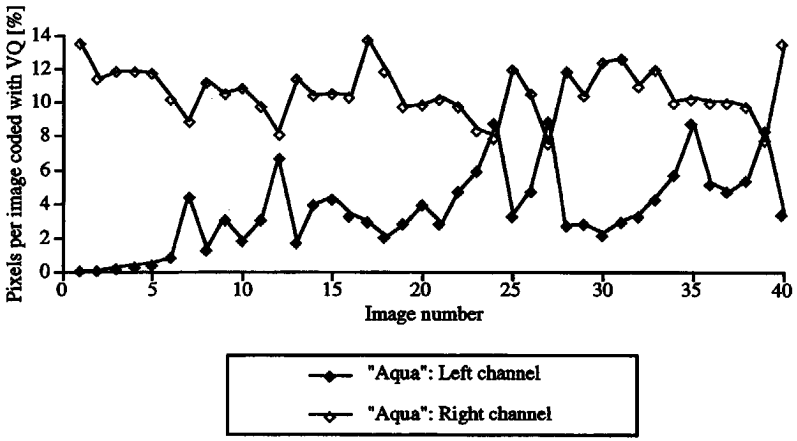


Figure 5.20: Percentage of pixels per image to be handled as a synthesis error in "Aqua"

Obviously, the number of pixels to be transmitted as a synthesis error increases when the PSNR decreases and is higher for the right images than for the left. What can be observed is that the curve for the right channel exhibits almost the same behaviour as the curve for the left channel, but with a delay of one image. This is due to the sequence of analysis as described in Section 4.4.3. The right image R_i is analysed at the same time as the left image L_{i-1}. Consequently, a high number of errors in L_{i-1} will result in a high number of errors in R_i as well.

When the coding principle for synthesis error described in Section 5.3 is used, the number of pixels in Figure 5.20 gives the number of bits required to code the synthesis error as shown in Figure 5.21.

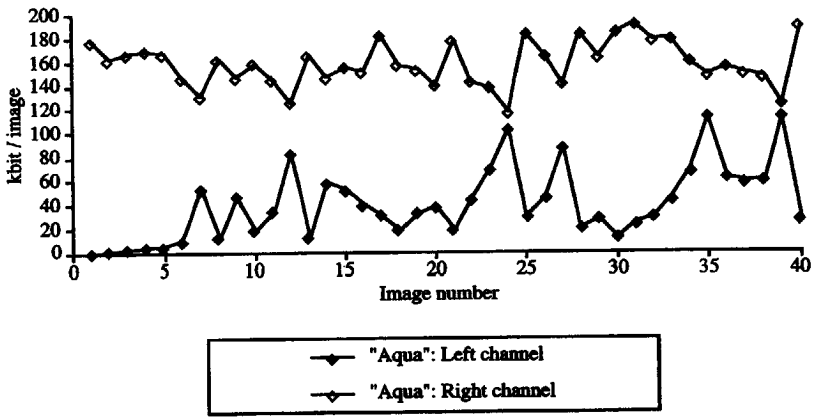


Figure 5.21: Bit rate per image required to transmit the synthesis error for "Aqua"

The bit rate required to transmit the synthesis error for the left channel is 1.6 Mbit for all 40 images. This is approx. 1 Mbit/s. For the synthesis error of the right channel, 3.9 Mbit/s are necessary. When the bit rate of 600 kbit/s for the parameter description of the regions is included, this means that a total bit rate of approx. 5.5 Mbit/s is required to encode the stereoscopic signal of "Aqua". However, the coding of the synthesis error is by no means perfect as it uses a pixel description for the error in a region-based coding scheme and also uses non-optimised code-books.

When the synthesis error is added to the predicted images, this gives the PSNR-values shown in Figure 5.22. Obviously, the PSNR-values are now higher compared with Figure 5.19 where no synthesis error was added. The PSNR-curve also looks smoother than before as single images were of a bad quality due to high errors.

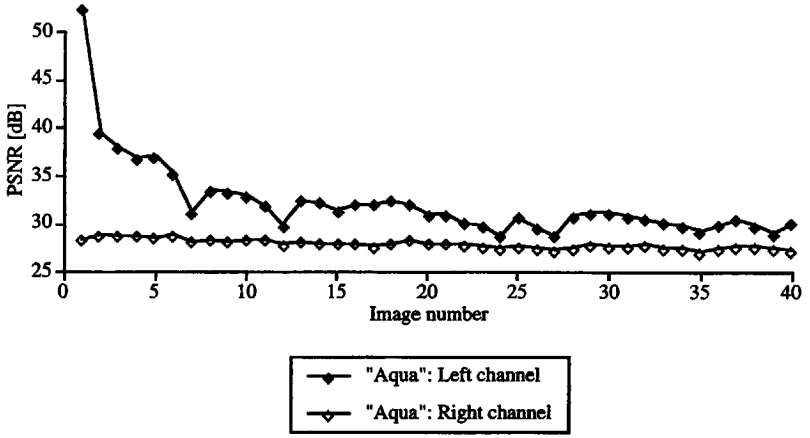


Figure 5.22: PSNR curve of "Aqua" after adding the synthesis error

The same experiments have been performed with the "Tunnel" sequence. Figure 5.23 shows the bits needed to transmit the region parameters.

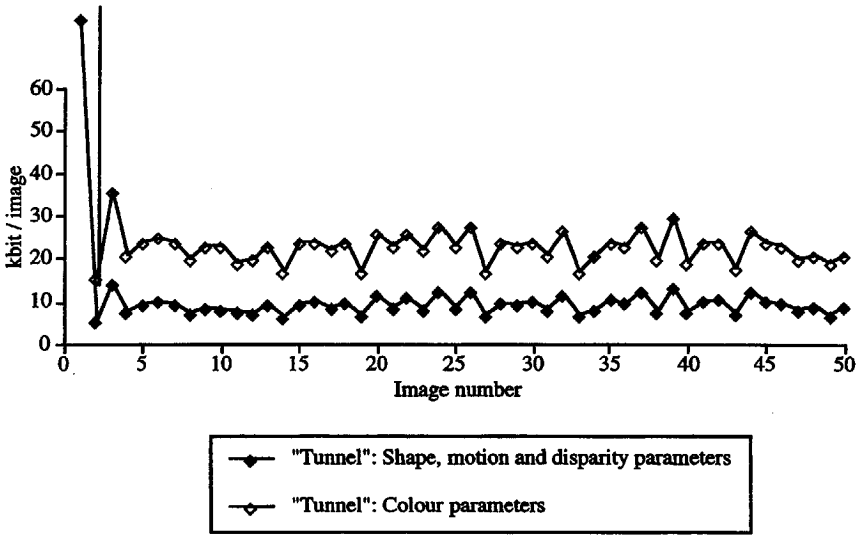


Figure 5.23: Bit rate for the region description of "Tunnel"

As with "Aqua", most bits are used to describe the colour parameters of the sequence: 1300 kbits were necessary for the 50 "Tunnel" images. An additional 530 kbits were used to

describe the motion, disparity and shape parameters. The total is 1830 kbits for the 50 images giving a bit rate of 915 kbit/s for the region parameters and the PSNR-values shown in Figure 5.24 when using only these parameters for the synthesis.

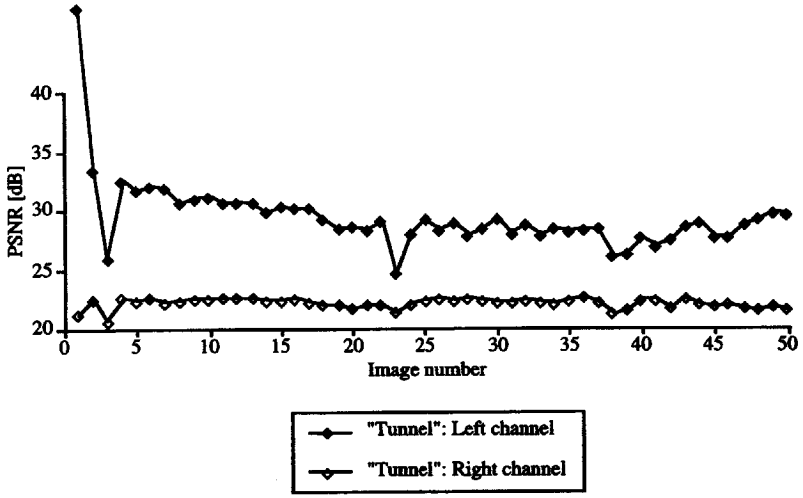


Figure 5.24: PSNR-values for synthesised "Tunnel" images using only the region parameters

As with "Aqua", the first left image of "Tunnel" is of very high quality, but the following images have a more constant quality than in "Aqua". The reason for this is, that in the "Tunnel" sequence there is no motion in large Sections of the images. The "Tunnel" sequence exhibits a stationary background and a train moving from right to left and from bottom to top. Since the unmoving background is transmitted only once in the first image, errors will mainly occur near the moving train. The size of these areas is approximately the same for the whole sequence and so only minor quality differences can be observed in the synthesised images. The exceptions are images number 3 and number 23, where a sudden quality decrease occurs. With these two images, the motion parameters could not be estimated precisely enough because of the limitations of the source model. Since the motion of the segmented regions of the train does not obey a simple translational relationship, using a motion model of this kind for "Tunnel" causes the incorrect positioning of the regions in most of the images. The use of rotational parameters could solve this problem.

For the right channel, the PSNR-values are lower again due to the combination of different disparity values in one region. Basically, the same course of values as for the left channel can be observed.

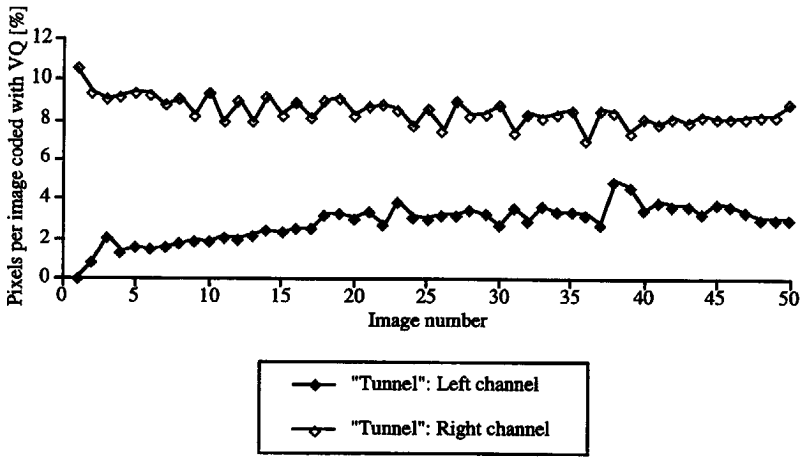


Figure 5.25: Percentage of pixels per image to be handled as a synthesis error in "Tunnel"

Figure 5.25 shows the percentage of pixels (based on the image size of 720x576 pixels) per image to be handled as a synthesis error as described in Section 5.3. Obviously, this curve corresponds very well to the PSNR curve in Figure 5.24. Where the PSNR is low, a high number of pixels have to be corrected, where the PSNR is high, a smaller number of pixels have to be corrected.

When vector quantisation is used, this gives the number of bits to be transmitted per image as shown in Figure 5.26.

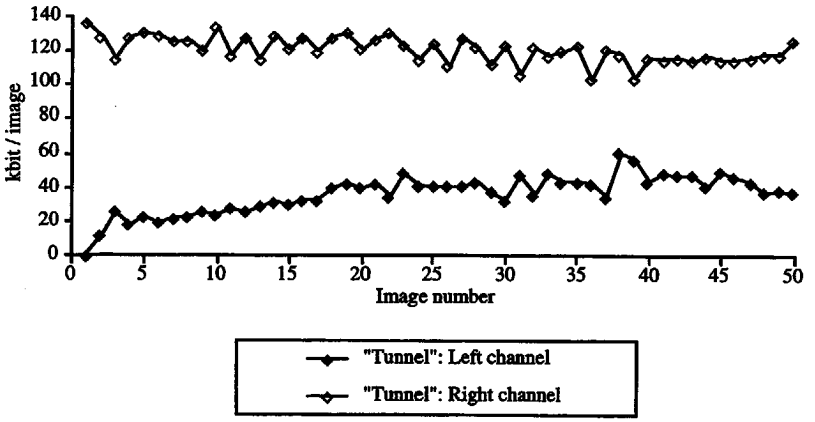


Figure 5.26: Bit rate per image required to transmit the synthesis error for "Tunnel"

Using this kind of coding for the synthesis error is again far from perfect. It uses far more bits than a highly sophisticated error coding method specially designed for the use in a region based coder would. The sum of the numbers in Figure 5.26 gives a data rate of 1.8 Mbit for the left channel and 6 Mbit for the right channel for all the 50 images. The time average in secs is 900 kbit/s for the left channel and 3 Mbit/s for the right channel. If the 900 kbit/s for the region parameters is added, a total bit rate of approx. 4.8 Mbit/s is obtained for the stereoscopic signal.

The final PSNR-values for the corrected "Tunnel" sequence are shown in Figure 5.27, where the quality is now quite high and the curve smoother than in previous graphs.

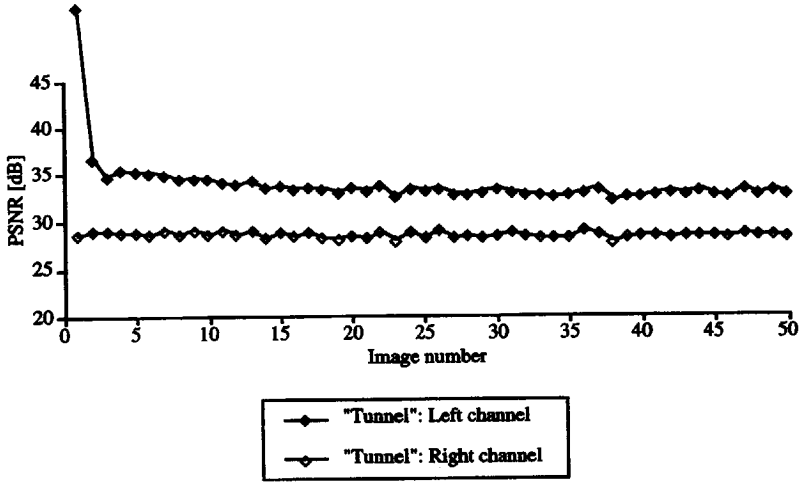


Figure 5.27: PSNR curve for "Tunnel" after adding the synthesis error

5.5.2 Informal Subjective Evaluation

Since the peak signal to noise ratio PSNR is calculated taking single pixel errors into account, small positioning and shape errors for the regions will decrease the PSNR. On the other hand, such errors will not necessarily reduce the visual quality of the images. In order to judge the subjective visual quality of the region-based coded images, a panel of experts - persons working on the coding of image sequences and so familiar with the drawbacks of block-based coding schemes - have been asked to judge the results visually. The evaluations were performed using a 3-DTV system as shown in Figure 5.28.

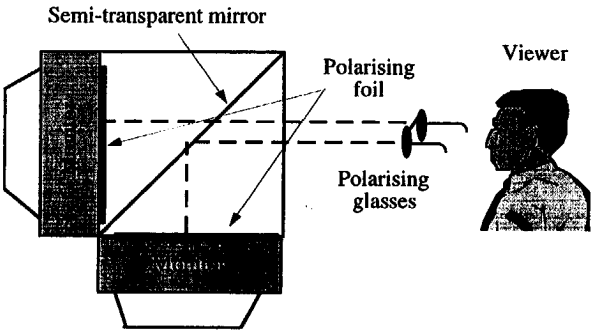


Figure 5.28: 3-DTV system used for subjective evaluations

In this system, two monitors are fixed vertically to each other. On the screen of each monitor is a polarising foil, the foils' polarisation being mutually perpendicular. A semi-transparent glass plate is fixed at an angle of 45° to each monitor. This allows a light beam from the top monitor through and it reflects a beam from the bottom monitor. Therefore, a viewer sees both images superimposed to each other. A pair of stereo glasses is made from polarising glass - one with a vertical polarisation vector and the other with a horizontal. With the stereo glasses, the viewer can have the two channels separately presented to each eye and can then see three dimensionally.

To compare the results with a standard coding scheme, both channels of the stereoscopic sequences "Aqua" and "Tunnel" have also been coded separately with a MPEG2 coder. The bit rate used for this experiment was half the bit rate used for the region-based coder for each of the two channels. In this way "Aqua" was coded with 2.75 Mbit/s for each of the two channels and "Tunnel" was coded with 2.4 Mbit/s per channel for comparison. When making their subjective evaluations, the viewers were asked to look at the stereoscopic images and judge the quality according to certain criteria.

As part of an initial experiment, individual stereoscopic images of "Aqua" and "Tunnel" were shown to the viewers. The synthesis error was not added to the region-based encoded images that were used. The viewers were asked whether they can see any artefacts. For "Aqua" a result produced by the region-based coder without adding the synthesis error can be seen in Figure 5.12. None of the viewers could see any artefacts. In the case of "Tunnel", artefacts were detected - particularly in the later images in the sequence where some of the regions in the background have been merged with the moving train, shifting the background to an incorrect position in the image. This can be seen in Figure 5.29 which shows one of the worst images of the sequence - in particular the last wagon of the train and the regions behind it.

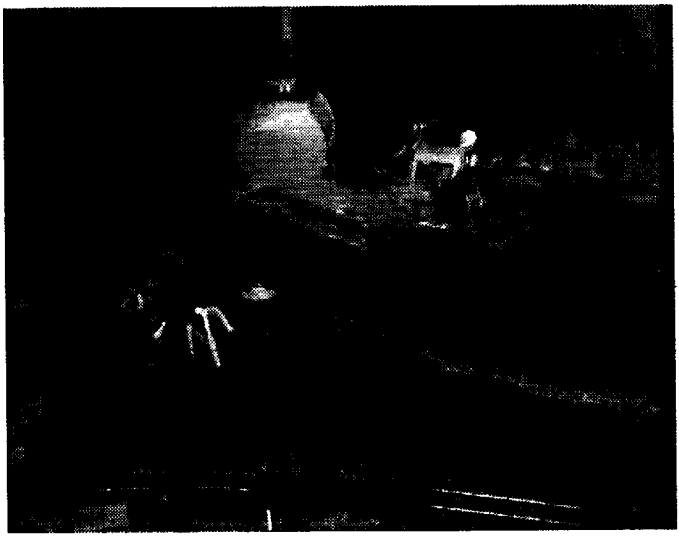


Figure 5.29: Synthesised right image 35 of "Tunnel" without adding the synthesis error

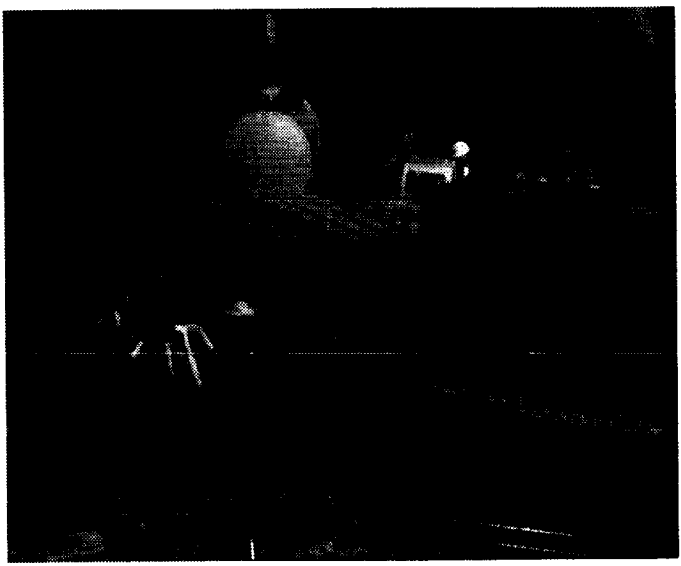


Figure 5.30: MPEG2-encoded right image 35 of "Tunnel"

When comparing these region-based encoded images with MPEG2-encoded images (see Figure 5.30), the judgement of the viewers was that the quality of the region-based encoded images was better or the same as the MPEG2-encoded ones. Bearing in mind that the two systems use rather different bit rates - a few hundreds of kbits and a couple of Mbits - this judgement shows the enormous potential of the region-based coder. In particular, the blurring in the MPEG2 B-frames, which are bi-directionally predicted and encoded at a rather low bit rate, disturbed the viewers (see Figure 5.31 (a)). This blurring did not occur in the region-based system (see Figure 5.31 (b)). Region contours were judged to be a lot sharper here, so enhancing the overall quality of the image.

(a) (b)

Figure 5.31: Illustration of the blurring effect with details from "Tunnel": (a) MPEG2 coder and (b) region-based coder (without adding the synthesis error)

With "Tunnel", even block-artefacts could be noticed in some of the MPEG2-encoded images (see Figure 5.33 (a)). Therefore, the overall judgement of all the viewers was that the region-based stereoscopic coder delivers better quality when looking at individual images of "Aqua". The region-based artefacts in some of the images of "Tunnel" were too intrusive and the viewers judged their quality as equal to that of the MPEG2-encoded images.

In the second experiment, the viewers were asked to look at the entire sequences and to judge their quality. Again, region-based encoded images without synthesis error addition were presented first. A couple of problems were identified in this way. The most serious artefacts occur in "Tunnel" whenever regions from the background are merged and shifted according to the motion of the train. A second problem occurs due to temporal inconsistencies in the motion vectors. Some of the regions, therefore, do not move smoothly but exhibit some jerkiness over time. Last but not least, there is no colour update for the regions in the sequences. The colour is only updated when the regions are merged and a new region description is transmitted. In such cases, the colour of some regions jumps from bright to

dark, which annoyed some of the viewers. All these problems occur occasionally in small sections of the image, but this was very annoying to the viewers nevertheless. If these sections are not taken into account when the quality of the images is assessed, the viewers judged the quality the same or better than MPEG2, where the bit rate was five or even ten times that of the region-based system without synthesis error. Again, the precise description of the region contours and the blurring of the images with MPEG2 were the reasons for this judgement.

When the synthesis error is added to the synthesised images, the major advantage of a region-based coder is lost. Now, the contours of the regions are no longer as sharp as they were when the error was not added. On the other hand, all the artefacts described above can no longer be seen (see Figure 5.32). When these corrected sequences were compared with an MPEG2-encoded stereoscopic signal (now using approximately the same bit rate for both coding methods) the viewers said "Aqua" and the corrected sequences were of the same quality. With "Tunnel", the subjective visual quality of the region-based image sequence is even judged better than that of the MPEG2 encoded sequence, where block artefacts are visible (see Figure 5.33). Obviously, the visual quality of the region-based stereoscopic image sequence coder will improve when the synthesis error is coded more intelligently. Apparently, adding the error is only necessary for selected image parts. The current approach using a threshold is too simple as more bits are used than is necessary for visual quality.

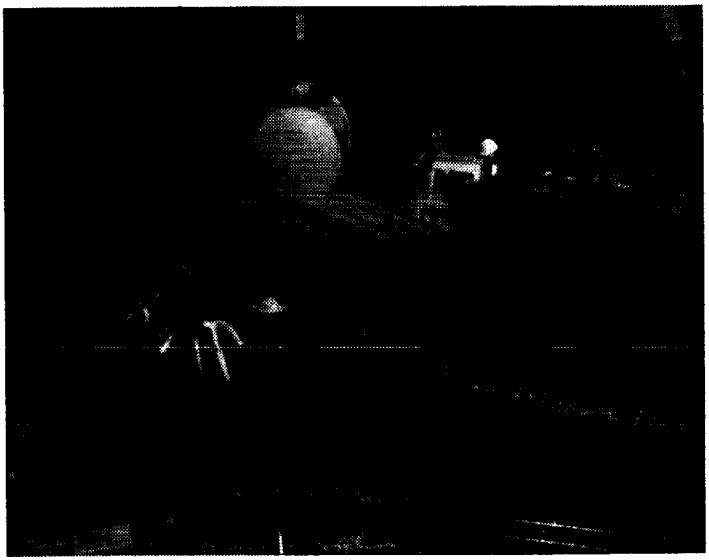
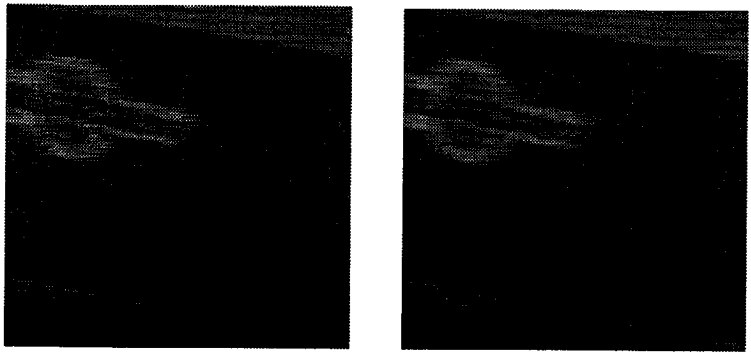


Figure 5.32: Synthesised right image 35 of "Tunnel" after adding the synthesis error



(a) (b)

Figure 5.33: Illustration of block artefacts with details from "Tunnel": (a) MPEG2 coder and (b) region-based coder (after adding the synthesis error)

5.6 Conclusion

In this Chapter, a region-based stereoscopic image-sequence coder making use of image analysis and synthesis methods has been presented. The coding of the region parameters was investigated as well as the possibility of coding the synthesis error. Also, the basic ideas underlying a rate control were described in this Chapter. Since the ways of influencing the bit rate are basically limited to quantisation of the colour parameters, the presented coder is basically a variable bit rate coder. Even with very basic coding of synthesis errors, the region-based coder delivers a subjective quality that is similar to that produced by an MPEG2 coder when the two stereoscopic channels of a sequence are encoded separately with the same total bit rate - as was shown in the subjective evaluation.

A region-based coder obviously behaves differently from a block-based coder. In a region-based encoded sequence, the boundaries of regions are well-defined and the regions themselves are of high quality, whereas the use of fixed blocks results in a blurred image. In this way, the images delivered by the region-based approach are much clearer and look more pleasant to the viewers. The viewers, therefore, accept other artefacts, say abrupt colour changes of the regions, and still state that the quality is better than that of MPEG2-encoded sequences.

Several inconsistencies with the temporal behaviour of the region-based coder were identified by the subjective evaluation. These inconsistencies are caused by the incorrect merging of

regions and the temporal jerkiness of regions due to incorrect motion vectors and the sudden changes of the region colour, due to missing updates of the region parameters over time.

When the synthesis error is coded as described, the advantage of having a clearer image becomes less important in the case of the region-based coder, but, on the other hand, other serious artefacts can be corrected with this type of error coding. However, a better coding of synthesis errors, which takes the regions into account and does not simply transmit error values when a threshold is exceeded, will probably decrease the bit rate without decreasing visual quality.

All in all, it has been demonstrated that the concept of the region-based coder can handle stereoscopic image sequences without knowledge of the epipolar geometry and the content of the scene and that it can deliver visual quality comparable with that of MPEG2 encoding.

6. Discussion

In this thesis a region-based stereoscopic image sequence coder has been developed. The coder is based on the analysis of the stereoscopic images applying region segmentation based on disparity and motion information. Prediction of the images is performed by image synthesis based on the region parameters stored in a region memory. Regions in the left images will be shifted according to their motion parameters; regions in the right images of the stereoscopic sequence will be shifted according to their disparity as well. The use of disparity is also a big advantage when applying motion compensation to the left image sequence. With the knowledge of the disparity of the regions - which is proportional to their distance from the camera - it is easy to decide which region is visible and which is covered. In this way, ambiguities which would lead to errors without the use of disparity can be resolved efficiently.

This Chapter discusses the advantages and disadvantages of a region-based coder of this kind compared with available, standard block-based coders such as MPEG2. Furthermore, it contains directions for future developments and improvements of the system.

6.1 Comparison of the Region-Based Coder with Block-Based Coders

The disadvantage of block-based coders with approximately constant quantisation is that all the colour parameters for large areas of the image have to be coded and transmitted, which results in a high bit-rate. With low-bit-rate, block-based coding, coarse quantisation is used to decrease the bit-rate accordingly. As a result, there is a reduction in the spatial resolution and coding errors are introduced.

Mosquito-artefacts are introduced because having only one displacement vector valid for the whole block is the strategy adopted for block-based coding. When this block is not completely within one region, but also covers part of a neighbouring region, errors will be produced. This residual error after displacement compensation then has to be coded and transmitted, which - particularly in the case of low-bit-rate-coding - is only possible with information loss. At the boundaries of the regions, visible coding errors will remain.

Apart from these mosquito artefacts, block-artefacts too will reduce the image quality in all areas. These artefacts occur because all displacement compensation errors- irrespective of their significance for a human viewer - will be updated with colour information. This means that small positioning-errors will also be corrected. As a human viewer concentrates on a natural displacement, but not on the exact positioning of a region in the image, these small positioning- and shape-errors are not really important to the human viewer [Höt92]. In block-based coders, therefore, a large number of bits are used to code areas that are not relevant. An advantage of region-based coders in this respect is possibility of spending bits in relevant areas to deliver good quality for the viewer.

Another important advantage of a region-based coder over block-based coders is a region-based content manipulation of the images as suggested by the MPEG4 standardisation group. Obviously, for the system that is being described, an interactive step would be necessary to combine several regions to form one real object as humans would "see" it. Nevertheless, this is an avenue that is closed to block-based coders as it would normally be impossible to combine a couple of equally sized square blocks to form an arbitrary object.

6.2 Future Work

As far as the region-based coder described in this thesis is concerned, one goal is not to transmit a synthesis error but to achieve a very high subjective quality without even general error coding. To increase the visual quality of the synthesised images, the artefacts that are familiar from the subjective tests have to be avoided.

The main artefacts, incorrect merging and jerkiness of regions, come from an inaccurate motion estimation. An improvement of the disparity and motion estimation will not only avoid these artefacts, but it will also contribute towards better segmentation and coding efficiency. Possible improvements include a subpixel accurate region-based disparity and motion estimation and the use of a more sophisticated source model.

Due to the horizontal ordering constraint, disparity and motion from one region can leak into a neighbouring one. One solution to this problem may be to partition the image into regions bounded by luminance edges and then apply dynamic programming to each region. On each line, the point where an edge crosses the scan line can be defined as a region boundary and dynamic programming can be applied locally between the two region boundaries. This may be a further improvement on the solution used for motion estimation where the luminance edge strength is included in the cost function.

Except for translational motion, there are other types of motion that are better described by affine transformations [AW93]. With an affine transformation, region manipulation will become more flexible. In this way, scaling (detected by the change of depth) and rotation can be handled. Moreover, by calculating the coefficients of the affine model for each region, they can then be used to create a cumulative segmentation which will progressively improve a segmented region [AW93].

Another important aspect of motion and disparity estimation is the need for temporally consistent vectors. Whenever vector "jumps" can be avoided, jerkiness of regions will no longer occur. A way to guarantee temporal consistency of vectors is a postprocessing step where the vectors for each region are followed over time and changed according to the values of the preceding and following region vectors.

Another way of improving the visual quality of the synthesised images is a regular update of the region colour parameters. In this way, abrupt changes of the colour values can be avoided; the colour would then be changed smoothly. Since the transmission of the colour differences will then be necessary for every image, this would obviously make it necessary to transmit the synthesised images at a higher data rate. On the other hand, error coding and transmission would be reduced or may not even be necessary at all.

When considering aspects relating to networks, network failure or cell loss must also be taken into account. In such cases, some kind of starting points in the bit stream have to be defined if the transmitter and the receiver are to be resynchronized. One way of doing this would be the introduction of a kind of I-frame as they are referred to in MPEG terminology. With a region-based coder, some of the images should be transmitted without using any kind of prediction. All the image analysis and synthesis should be performed again starting with an image totally defined as uncovered background. In this way, the coder would be forced to segment and transmit all the parameters again. If there was a network failure, these images could then be used as a new starting point.

Apart from these algorithmic questions, further work will include investigations into the hardware feasibility and the possible applications of the region-based stereoscopic coder. As the system that has been described is very complex and involves a large number of different steps to synthesise stereoscopic video, dedicated hardware, which is not yet available, is necessary to implement a real-time system. One of the most promising components that will soon be realised as hardware is disparity estimation. It is based on a dynamic programming approach estimating the vectors independently for each line of the image. It can, therefore, be implemented easily in parallel. As far as image analysis and synthesis are concerned, activities associated with the standardisation of MPEG4 are pushing forward the development of dedicated hardware.

Consequently, the first announcements of accelerator boards, or even VLSI implementations, for image analysis and synthesis can be expected within the next couple of years.

For this reason, applications in the near future will be restricted to systems based on software that do not operate in real time. Professional applications in industry or medicine, such as planning surgical operations, air traffic control systems, video archives or CAD, could benefit greatly from region-based stereoscopic systems of this kind. If some application-specific modifications are made to the system, these applications will be the first to make use of the additional features not provided by the standard block-based system which is available now.

References

[Adi85] G. Adiv
"Determing three-dimensional motion and structure form optical flow generated by several moving objects"
IEEE Pattern Analysis Machine Intelligence, Vol. PAMI-7, No. 4, July 1985, pp. 384-401

[AHS89] K. Aizawa, H. Harashima , T. Saito
"Model-Based Analysis-Synthesis Image Coding System for a Person's Face"
Signal Processing: Image Communications, Vol. 1, No. 2, October 1989, pp. 139-152

[ANG95] M. Accame, F.G.B. De Natale, D. Giusto
"Hierarchical block matching for disparity estimation in stereo sequences"
International Conference on Image Processing, 23-26 October 1995, Washington, DC, USA

[AW93] E. Adelson, J. Wang
"Representing moving images with layers"
MIT Media Lab, Perceptual computing group, Technical Report No. 228, April 1993

[BB81] H. Baker, T. Binford
"Depth from edge and intensity based stereo"
Proceedings IJCAI (1981), Vancouver, Canada

[BDP95] K. Bhalla,N. Durdle, A. Peterson, J. Raso, D. Hill, X. Li
"Automatic feature detection and correspondence in a stereo-vision application"
1995 IEEE International Conference on Systems, Man and Cybernetics. Vancouver, BC, Canada, 22-25 Oct. 1995

[Bel57] R. Bellman
"Dynamic Programming"
Princeton University Press, 1957

[BK95] P. Brigger, M. Kunt
"Morphological contour coding using structuring functions optimized by genetic algorithms"
ICIP-95, Washington DC, USA, 23-26 October 1995, pp. 534-537

[BN96] D. Bhat, S. Nayar
"Ordinal measures for visual correspondence"
Proceedings 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 18-20 June 1996

[CCL95] K. Chow, K. Chan, M. Liou
"Object-based coding method for visual telephony using discrete wavelet transform"
Proceedings of the SPIE, Taipei, Taiwan, 24-26 May 1995

[CDP95] B. Choquet, J.L. Dugelay, D. Pele
"A coding scheme for stereoscopic television sequences based on motion estimation -compensation using a 3-D approach"
5th International Conference on Image Processing and its Applications, 4-6 July 1995, London, UK

[Cha95] F. Chassaing
"Digital compatible transmission of a stereoscopic Television signal"
International Workshop on Stereoscopic and Three Dimensional Imaging, 6-8 September 1995, Santorini, Greece

[CJB94] I. Corset, S. Jeannin, L. Bouchard
"MPEG4: Very low Bit Rate Coding for Multimedia Applications"
SPIE Proceedings Vol. 2308
25-28 Sept. 1994, Chicago, USA

[CL95] L. Cinque, L. Lombardi
"Shape description and recognition by a multiresolution approach"
Image and Vision Computing (1995), Vol. 13, No. 8, pp. 599-607

[Cla85] R.J. Clarke
"Transform Coding of Image"
©1985 Academic Press

[Dan75] S. Danø
"Nonlinear and dynamic programming"
© 1975 Springer Verlag

[DC95] A. Delopoulos, A. Constantinides
"Object oriented motion and deformation estimation using composite segmentation"
ICIP-95, 23-26 October 1995, Washington DC, USA

[DIS92] RACE project DISTIMA
"Selection of Short Existing sequences"
Deliverable R2045/CCETT/WP3.1/DS/T/004, July 1992

[DIS94] RACE project DISTIMA
"Production of the first experimental normalized sequences"
Deliverable R2045/CCETT/WP3.1/DS/T/024-28, February 1994

[DK95] V. Delport, M. Koschorreck
"Genetic algorithm for codebook design in vector quantisation"
Electronics Letters (1995) vol.31, no.2, p.84-5. 4

[EF77] P. Eckman, V.W. Friesen
"Facial Coding System"
Consulting Psychologist Press Inc., Palo Alto, 1977

[EK85] M. Eden, M. Kocher
"On the performance of a contour coding algorithm in the context of image coding. Part 1: Contour segment coding"
Signal Processing, vol. 8, no. 4, July 1984, pp. 381-386

[Fal94] L. Falkenhagen
"Depth Estimation from Stereoscopic Image Pairs Assuming Piecewise Continuous Surfaces"
European Workshop on combined real and synthetic image processing for broadcast and video productions, 23-24 November 1994, Hamburg, Germany

[FLB94] R.E.H. Franich, R.L. Lagendijk, J. Biemond
"Fractal coding in an object-based system"
ICIP-94, 13-16 November 1994, Austin, TX, USA

[FLB94a] R.E.H. Franich, R.L. Lagendijk, J. Biemond
"Object Based Stereoscopic Coding: Vector Field Estimation and Object Segmentation"
EUSIPCO-94, Lausanne, Switzerland

[Fra89] U. Franke
"Regionenorientierte Bildbeschreibung - Algorithmen und Möglichkeiten"
PhD Thesis, RWTH Aachen
Fortschritt-Berichte VDI, Reihe 10, Nr. 101, 1989

[Fra92] R.E.H. Franich
"Balance Compensation for Stereoscopic Sequences"
RACE project DISTIMA
Discussion Note R2045/TUD/WP3.2/DN/C/27.7.92/1, July 1992

[Fra96] R.E.H. Franich
"Disparity Estimation in Stereosocpic Digital Images"
PhD Thesis, Technical University Delft, 1996

[FS95[L. Falkenhagen, M. Strintzis
"3-D Object-Based Depth Estimation from Stereoscopic Image Sequences"
RACE DISTIMA Deliverable R2045/UH/DS/P/039/b1, March 1995

[GB88] B. Gillam, E. Borsting
"The role of monocular regions in stereoscopic displays"
Perception, Vol. 17, 1988, pp. 603-608

[GEM89] M. Gilge, T. Engelhart, R. Mehlan
"Coding of arbitrarily shaped image segments based on a generalized orthogonal transform"
Signal Processing: Image Communication, Vol. 1, 1989, pp. 153-180

[GK87] W. Geuen, F. Kappei
"Principle strategy of model based source coding"
Picture Coding Symposium (PCS'87), June 1987, Stockholm, Sweden

[GLY95] D. Geiger, B. Ladendorf, A. Yuille
"Occlusions and binocular stereo"
Int. Journal on Computer Vision (1995), Vol.14, No.3, pp. 211-226

[GSF95] B. Girod, E. Steinbach, N. Färber
"Comparison of the H.263 and H.261 Video Compression Standards"
SPIE Proceedings Vol. CR60
Standards and Common Interfaces for Video Information Systems, Oct. 25-26 1995, Philadelphia, USA

[Hor86] B. K. P. Horn
"Robot Vision, Chapter 13: Photogrammetry & Stereo"
©1986 MIT Press, Cambridge, MA, USA

[Hor93] R. ter Horst
"A demonstrator for coding and transmission of stereoscopic video signals"
4th European Workshop on Three-Dimensional Television, October 1993, Rome, Italy

[Höt90] M. Hötter
"Object-Oriented analysis-synthesis coding based on moving two-dimensional objects"
Picture Coding Symposium (PCS'90), March 1990, Cambridge, MA, USA

[Höt90a] M. Hötter
"Predictive contour coding for an object-oriented analysis-synthesis coder"
IEEE International Symposium on Information Theory, January 1990, San Diego, CA, USA

[Höt92] M. Hötter
"Objektorientierte Analyse-Synthese-Codierung basierend auf dem Modell bewegter, zweidimensionaler Objekte"
PhD Thesis, University of Hannover
Fortschritt-Berichte VDI, Reihe 10, Nr. 217, 1992

[IE94] E. Izquierdo, M. Ernst
"Motion/Disparity analysis and image synthesis for 3DTV"
International Workshop on HDTV (1994), 26-28 October 1994, Turin, Italy

[KA93] A. Kaup, T. Aach
"Region-based image coding using functional approximation"
Picture Coding Symposium (PCS'93), March 1993, Lausanne, Switzerland

[Kau95] A. Kaup
"Modelle zur regionenorientierten Bildbeschreibung"
PhD Thesis, RWTH Aachen
Fortschritt-Berichte VDI, Reihe 10, Nr. 381, 1995

[Kau97] A. Kaup
"On the Performance of the Shape Adaptive DCT in Object-Based Coding of Motion Compensated Difference Images"
to be published at Picture Coding Symposium (PCS'97), September 1997, Berlin, Germany

[KD94] K.S Kumar, U.B. Desai
"New algorithms for 3-D surface description from binocular stereo using integration"
Journal of the Franklin Institute (1994), Vol. 331B, No. 5, pp.531-554

[Kir89] H. Kirchner
"Ein mehrstufiger Ansatz zur Bewegungserkennung in Bildfolgen"
PhD Thesis, University of Erlangen-Nürnberg, 1992

[Koc91] R. Koch
"Adaptation of a 3D facial mask to human face in videophone sequences using model based image analysis"
Picture Coding Symposium (PCS'91), September 1991, Tokyo, Japan

[Koc83] M. Kocher
"Codage d'images à haute compression basé sur un modèle contour-texture"
Diss. Ecole Politechnique Fédérale de Lausanne, Nr. 476, 1983

[LC78] R. Larson, J. Casti
"Principles of Dynamic Programming"
©1978 Marcel Dekker Inc., New York

[Leo87] R. Leonardi
"Segmentation adaptive pour le codage d'images"
Diss. Ecole Politechnique Fédérale de Lausanne, Nr. 691, 1987

[Lin95] C.A. Lindley
"JPEG-like image compression"
Dr. Dobb's Journal (1995) vol.20, no.7, p.50, 52, 54-8.

[Liu95] J. Liu
"Stereo image compression - the importance of spatial resolution in half occluded areas"
SPIE Vol. 2411, February 1995, San Jose CA, USA, pp. 271-276

[LP96] C. Liu, A. Poularikas
"A new subband coding technique using (JPEG) discrete cosine transform for image compression"
Proceedings of the Twenty-Eighth Southeastern Symposium on System Theory, Baton Rouge, LA, USA, 31 March-2 April 1996

[Mat95] W. Mattern
"Audio and Video Compression with the MPEG standards"
Texas Instruments Technical Journal (1995), Vol. 12, no.6, pp 72-86

[MB94] F. Meyer, P. Bouthemy
"Region-based tracking using affine models in long image sequences"
CVGIP: Image Understanding (1994), Vol. 60, No. 2, pp. 119-140

[MHO89] H.G. Musmann, M. Hötter, J. Ostermann,
"Object-Oriented Analysis-Synthesis Coding of Moving Images",
Signal Processing: Image Communications, Vol. 1, No. 2, October 1989
pp.117-138

[MPEG90] MPEG Simulation Model Editorial Group
"MPEG Video Simulation Model Three (SM3)"
Doc. ISO/IEC JTC1/SC2/WG11/N0010

[MPEG96] Ad hoc Group on MPEG 4 video VM editing
"MPEG-4 Video Verification Model Version 3.0"
Doc. ISO/IEC JTC1/SC29/WG11/N1277

[MPEG97] MPEG 4 - Video Group
"Visual Working Draft 2.0"
Doc. ISO/IEC JTC1/SC29/WG11/N1583

[MMP96] J.R. Morros, F. Marqués, M. Pardàs
"Video Sequence Coding Based on Rate-Distortion Theory"
SPIE Vol. 2727, March 1996, Orlando FL, USA, pp. 1185-1196

[Mus95] H.G. Musmann
"A layered coding system for very low bit rate video encoding"
Signal Processing: Image Communications, Vol. 7, No. 4-6, pp. 267-278

[MVD96] X. Marichal, C. De Vleeschouwer, T. Delmot, B. Macq
"Object based coding through multigrid representation"
Proceedings of the SPIE, San Jose, CA, USA, 31 Jan.-2 Feb. 1996

[NH88] Arun N. Netravali, Barry G. Haskell,
"Digital Pictures"
©1988 AT&T Bell Laboratories, Plenum Publishing Corporation

[NHC91] Y. Nakaya, H. Harashima, Y.C. Chiang
"Model-based / MC-DCT hybrid coding system"
Picture Coding Symposium (PCS'91), September 1991, Tokyo, Japan

[Nie90] H. Niemann
"Pattern Analysis and Image Understanding"
© 1990 Springer Verlag, Series in Information Science

[OK85] Y. Ohta, T. Kanade
"Stereo by intra- and inter-scanline search using dynamic programming"
IEEE Transactions on Pattern Analysis and Machine Intelligence, March 1985, Vol. APMI-7, No. 2, pp. 139-154

[OS95] T. O'Rourke, R. Stevenson
"Human visual system based wavelet decomposition for image compression"
Journal of Visual Communication and Image Representation (1995), Vol. 6, No. 2, pp. 109-121

[Ost90] J. Ostermann
"Modelling of 3D-moving objects for an analysis-synthesis coder"
SPIE/SPSE Symposium on Sensing and Reconstruction of 3D Objects and Scenes '90, SPIE Proceedings, Vol. 1260, February 1990, Santa Clara, USA, pp. 240-250

[Pan1858] P. Panum
"Physiologische Untersuchungen über das Sehen mit zwei Augen"
Homann Verlag, Kiel, 1858

[PC96] S. Panis, J. Cosmas
"Motion Estimation with object-based regularisation"
IEE Electronis Letters, Vol. 32,. No. 10, pp. 872-873, May 1996

[PD96] D. Papdimitriou, T. Dennis
"Epipolar line estimation and rectification for stereo image pairs"
IEEE Transactions on Image Processing (1996), Vol. 5, No. 4, pp672-676

[PF77] E. Persoon, K.S. Fu
"Shape discrimination using fourier descriptors"
IEEE Trans. System Man Cybernetics (1977), Vol. SMC-7, pp. 170-179

[Phi96] W. Philips
"A comparison of four hybrid block/object image coders"
Signal Processing (1996) vol.54, no.1, p.103-107

[PKH95] A. Puri, R. Kollarits, B. Haskell
"Stereoscopic video compression using temporal scalability"
SPIE Proceedings, Vol. 2501, May 1995, Taipei, Taiwan

[PM93] W. Pennebaker, J. Mitchell
"JPEG Still Image Data Compression Standard"
©1993 van Nostrand Reinhold

[PN95] L. Peisuei, T. Nagao
"Hierarchical description of two dimensional shapes using a genetic algorithm"
IEEE Conference on Evolutionary Computation, 29 November - 1 December 1995, Perth WA, Australia,

[POA94] O. Parlaktuna, S. On, M. Akcay
"A feature based stereo matching algorithm"
7th Mediterranean Electrotechnical Conference, 12-14 April 1994, Antalya, Turkey

[Pot75] J.L. Potter
"Velocity as a cue to segmentation"
IEEE Transactions on Syst. Man. Cybern, May 1975, pp. 390 -394

[Pra91] W. Pratt
"Digital Image Processing"
© 1991 John Wiley & Sons

[PS94] M. Pardas, P. Salembier
"Time-recursive segmentation of image sequences"
Proceedings of EUSIPCO-94, Edinburgh, UK, 13-16 Sept. 1994

[PZC95] S. Panis, M. Ziegler, J. Cosmas
"A system approach to Disparity Estimation"
IEE Electronis Letters, Vol. 31,. No. 11, pp. 871-873, May 1995

[PZC97] S. Panis, M. Ziegler, J. Cosmas
"The Use of Stereo and Motion in a Generic Object Based Coder"
Signal Processing: Image Communication (1997) vol.9, p.221-238

[QCH94] R. De Queiroz, C. Choi, Y. Huh, Hwang, K. Rao
"Wavelet transforms in a JPEG-like image coder"
Proceedings of the SPIE, Chicago, IL, USA, 25-28 Sept. 1994

[RG96] F. Rannou, J. Gregor
"Equilateral polygon approximation of closed contours"
Pattern Recoginition (1996), Vol. 29, No. 7, pp. 1105-1115

[RH96] K.R. Rao, J.J. Hwang
"Techniques and Standards for Image, Video and Audio Coding"
©1996 Prentice-Hall Inc.

[SBM95] T. Sikora, S. Bauer, B. Makai
"Efficiency of shape-adaptive 2-D transforms for coding of arbitrarily shaped image segments"
IEEE Transactions on Circuits and Systems for Video Technology (1995) vol.5, no.3, p.254-258

[Sch95] K. Schröder
"Compact description of objects for object-based moving-image coding"
ITG-Fachberichte No. 136, Dortmund, Germany, 4-6 October 1995, pp. 209-214

[Sha93] J. Shapiro
"Embedded image coding using zerotrees of wavelet coefficients"
IEEE Transactions on Signal Processing (1993), Vol. 41, No. 12, pp.3445-3462

[Sik96] T. Sikora
"Low complexity shape-adaptive DCT for coding of arbitrarily shaped image segments"
Signal Processing: Image Communication (1996) vol.7, no.4-6, p.381-395

[SIM94] COST 211 Simulation Group Report SIM(94)61, "SIMOC1"

[SM95] T. Sikora, B. Makai
"Shape-Adaptive DCT for generic coding of video"
IEEE Transactions on Circuits and Systems for Video Technology, Vol. 5, No. 1, February 1995

[SSG96] K. Stuhlmüller, A. Salai, B. Girod
"Rate-constrained contour-representation for region-based motion compensation"
SPIE Vol. 2727, Orlando, FL, USA, 17-20 March 1996, pp. 344-355

[SSJ95] S. Sethuraman, M.W. Siegel, A.G. Jordan
"A multiresolutional region based segmentation scheme for stereoscopic image compression"
SPIE Proceedings Vol. 2419
5-11 February 1995, San Jose, CA, USA

[Ste93] A. Stein
"Untersuchungen zur Konturcodierung im Rahmen der regionenorientierten Bildbeschreibung"
Diploma thesis, RWTH Aachen, January 1993

[SV93] R. Schäfer, J. Vlontzos
"Specifications of Demonstrator"
Deliverable R2045/IRT/DS/I/013/b1, February 1993

[TA94] B. Tseng, D. Anastassiou
"Compatible video coding of stereoscopic sequences using MPEG2 scalability and interlaced structure"
6th International Workshop on HDTV, 26-28 October 1994, Torino, Italy

[TA95] B. Tseng, D. Anastassiou
"Perceptual adaptive quantization of stereoscopic video coding using MPEG2's temporal scalability structure"
International Workshop on Stereoscopic and Three Dimensional Imaging, 6-8 September 1995, Santorini, Greece

[TAB95] A.M. Tekalp, Y. Altunbasa, G. Bozdagi
"Two- versus three-dimensional object-based coding"
SPIE Proceedings Vol.2501
24 - 26 May, 1995, Taipei, Taiwan

[TGS95] D. Tzovaras, N. Grammalidis, M. Strintzis
"Object-based coding of stereo image sequences using joint 3D motion/disparity segmentation"
SPIE Proceedings Vol.2501
24 - 26 May, 1995, Taipei, Taiwan

[TGS96] D. Tzovaras, N. Grammalidis, M. Strintzis
"Joint three-dimensional motion/disparity segmentation for object based stereo image sequence coding"
Optical Engineering (1996), Vol. 35, No. 1, pp 137-144

[TW96] M. Turner, N. Wiseman
"Efficient lossless image contour coding"
Computer Graphics Forum (1996) vol.15, no.2, p.107-117

[Ull79] S. Ullman
"The Interpretation of Visual Motion"
MIT Press, Cambridge, MA, 1979

[Wal91] M. Waldowski
"A new segmentation algorithm for videophone applications based on stereo image pairs"
IEEE Transactions on Communications (1991), Vol. 39, No. 12, pp. 1856-1868

[Woo91] J. Woods
"Subband Image Coding"
©1991 Kluwer Academic Publishers

[WZJ96] H. Wang, T. Zhuang, D. Jiang, W. Liu, I. Magnin, G. Gimenez
"Improved adaptive boundary tracing using 2D dynamic programming"
SPIE Vol. 2727, Orlando, FL, USA, 17-20 March 1996, pp. 199-208

[YC94] Y. Yang, Y. Cheng
"MPEG-based coding algorithm for 3D-TV"
6th International Workshop on HDTV, 26-28 October 1994, Torino, Italy

[YGB90] A. Yuille, D. Geiger, H. Bulthoff
"Stereo, mean field theory and psychophysics"
1st European Conference on Computer Vision, April 1990, Antibes, France

[YML95] P. Yuen, S. Ma, J. Liu et al
"Contour decomposition using dominant points and moment difference"
Image Analysis Applications and Computer Graphics (ICSC 95), 11-13 December 1995, Hong Kong

[Zie92] M. Ziegler
"Disparity Estimation Using Variable Blocksize"
3rd European Workshop on 3DTV, November 1992, Rennes, FRANCE

[Zie92a] M.Ziegler et. al.
"Digital Stereoscopic Imaging & Applications, A Way towards new dimensions. The RACE II project DISTIMA"
IEE Colloquium on Stereoscopic Television, October 1992, London, UK

[ZP93] M. Ziegler, S. Panis
"Object based stereoscopic image coding"
International Workshop on HDTV (1993), October 1993, Ottawa, Canada

[ZP96] M. Ziegler, S. Panis
"Object-oriented coding of stereoscopic sequences"
1996 Picture Coding Symposium, March 1996, Melbourne, Australia

[ZT92] M. Ziegler, W. Tengler
"Stereo Compensated Coding"
International Symposium on three dimensional Image Technology and Arts, February 1992, Tokyo, Japan

[ZTS94] M. Ziegler, W. Tengler, A. Starck
"Coding scheme and hardware structure of a high-rate digital HDTV codec with partly error-free encoding"
Signal Processing: Image Communications 6 (1994), pp. 163-172

[ZTT91] M. Ziegler, W. Tengler, P. Tabeling
"Influence of Camera Calibration on the Coding of Stereo Sequences"
First International Festival on 3-D Images, September 1991, Paris, France

Gebiedsgeoriënteerde Analyse en Codering van Stereoscopische Beeldreeksen

Samenvatting

De mens kan de wereld driedimensionaal zien dankzij het binoculaire gezichtsvermogen. Diepte kan worden "gezien" omdat er kleine verschillen zijn in de beelden die het linker- en rechteroog ontvangen. Bij computer vision nemen twee camera's de plaats in van de menselijke ogen, en simuleren op deze wijze het natuurlijke gezichtsvermogen. Dergelijke technische stereoscopische systemen zullen in toekomstige toepassingen op het gebied van de telecommunicatie, zoals telepresentatie, een steeds belangrijkere rol gaan spelen.

De beschikbare bandbreedte voor de transmissie van videobeeldsignalen is beperkt. Daarom zal het nodig zijn de bitsnelheid te verlagen terwijl de beeldkwaliteit op aanvaardbaar niveau blijft. De bitsnelheid wordt door datacompressietechnieken verlaagd waarbij enerzijds getracht wordt de benodigde grote bandbreedte en daarmee de transmissiekosten te minimaliseren, maar waarbij anderzijds aanvaardbare reconstructie van het beeld gewaarborgd blijft. In het geval van stereoscopische beelden wordt de oorspronkelijke bitsnelheid verdubbeld, aangezien er twee beelden worden verzonden in plaats van één. Dit maakt datacompressie nog meer noodzakelijk.

Bij beelden die met twee camera's verkregen zijn, is er sprake van een verschuiving van informatie tussen het linker en rechter spatiële beeld van een stereoscopisch beeldenpaar. Deze verschuiving, ook wel *dispariteit* genoemd, is een uniek fenomeen voor stereoscopische beelden. Omdat dispariteit omgekeerd evenredig is met de diepte, kan het bij de analyse van stereoscopische beeldenparen worden gebruikt. In dit proefschrift wordt een nieuwe *coderingstechniek voor stereoscopische beeldreeksen* voorgesteld die gebruikt maakt van dispariteit. Het onderzoek dat tot dit proefschrift heeft geleid werd voornamelijk verricht in het kader van het Europese project DISTIMA (DIgital STereoscopic IMaging & Applications).

Gedurende de laatste jaren heeft objectgeoriënteerde codering -een nieuw coderingsconcept- wereldwijd veel aandacht gekregen. Door vorm, beweging en kleur van de objecten in een beeld te verzenden, kunnen storende coderingsfouten ("mosquito" effecten, "blocking" artefacten) zoals die optreden bij blokgeoriënteerde hybride codering worden vermeden. Bovendien kunnen belangrijke beeldgebieden, zoals bijvoorbeeld de details van een gezicht in een teleconferentie-situatie, met een hogere beeldkwaliteit worden gereconstrueerd dan met blokgeoriënteerde

hybride codering. Bewegende objecten vormen echter een groot probleem. Als twee afzonderlijke objecten naar elkaar toe bewegen, is het bij objectgeoriënteerde coderingstechnieken meestal niet bekend welk object zich voor het andere bevindt. Dit levert een grote coderingsfout op als bij de reconstructie van het beeld het verkeerde object wordt gekozen. Stereoscopische informatie verhelpt deze beperking van objectgeoriënteerde codering. Met behulp van diepte-informatie (geschat uit het stereoscopische signaal), kan gemakkelijk worden vastgesteld welk object het dichtste bij de camera en dus zichtbaar, is. Daarom is het aan te raden niet alleen de drie parameters: vorm, kleur en beweging te coderen en te verzenden, maar deze te laten vergezellen van de vierde parameter: diepte of dispariteit.

Met behulp van dispariteit kan de positie van een object in de ruimte worden vastgesteld. Echter, omdat de in dit proefschrift ontwikkelde coderingstechniek geen echte fysieke objecten gebruikt maar beeldgebieden die een zekere positie in de ruimte hebben maar niet noodzakelijkerwijs met fysieke objecten overeenkomen, spreken we in dit proefschrift van een *gebiedsgeoriënteerde* codeertechniek.

Dit proefschrift beschrijft een *gebiedsgeoriënteerde stereoscopische coderingstechniek voor beeldreeksen*, die op de principes van beeldanalyse en -synthese is gebaseerd. Het bronmodel veronderstelt gebieden van willekeurige, maar wel vaste vorm die een translatorische beweging maken in plaats van een vaste opdeling van het beeld in blokken van bijvoorbeeld 8x8 beeldpunten. Aan de zenderzijde worden deze gebieden gevonden door middel van segmentatie waarbij gebruik wordt gemaakt van bewegings- en dispariteitsvectoren. Elk gebied wordt vervolgens gekarakteriseerd door een set parameters (inclusief kleur, vorm, beweging en dispariteit). Diverse beeldanalysetechnieken worden ingezet om de parameters te bepalen. Voor het verlagen van de bitsnelheid worden de parameters vervolgens gecodeerd met behulp van standaard-coderingstechnieken, zoals temporele en spatiele predictie en entropie-codering. Aan de ontvangerzijde wordt gebruik gemaakt van beeldsynthese om uit de gecodeerde parameters een beeld te reconstrueren en het overeenkomende ruimtelijke beeld in een stereoscopisch beeldenpaar.

De belangrijkste ontwikkelingen in dit proefschrift zijn:

- Een nauwkeurige *dispariteits- en bewegingsschatter* (op één beeldpunt nauwkeurig). Dit is nodig voor het segmenteren van de gebieden. In dit proefschrift wordt een dispariteitsschatter ontwikkeld die zonder voorkennis van stereoscopische geometrie werkt en derhalve met nagenoeg elk stereoscopisch beeldenpaar kan omgaan. De voorgestelde method voert een optimalisatie uit op basis van dynamisch programmeren. Een optimalisatiecriterium wordt voorgesteld dat gebruik maakt van intensiteitsverschillen tussen beeldpunten in het stereoscopische beeldpaar, uitgaande van een het vloeiend verloop van de dispariteit binnen gebieden, en dat tevens rekening houdt met dispariteitssprongen en -occlusies op de grenzen van verschillende gebieden. Uit de analyse blijkt dat de hieruit voortvloeiende vectorvelden

tot op één beeldpunt nauwkeurig zijn, en van hoge kwaliteit. Deze velden vormen derhalve een goede basis voor de verdere verwerking zoals de segmentatie van de gebieden volgens hun dispariteit. De experimenten in dit proefschrift laten zien dat waarbij dynamisch programmeren niet alleen op dispariteitsschatting kan worden toegepast, maar ook de toepassing op bewegingsschatting van dezelfde methode resulteert in een hoogwaardig bewegingsvectorveld.

- *Beeldanalyse* van het linker beeld uit het stereoscopische paar gebaseerd op bovengenoemde dispariteits- en bewegingsvectorvelden. Allereerst worden initiele gebieden gesegmenteerd overeenkomstig het bronmodel. De segmentatie is uitsluitend gebaseerd op de dispariteits- en bewegingsvectorvelden. Nadat door samenvoeging een geschikt aantal gebieden is verkregen, worden voor elk van deze gebieden de vier vereiste parameters (dispariteit, beweging, vorm en kleur) berekend. De evaluatie laat de goede kwaliteit zien van de segmentatie van het beeld en de daaropvolgende beschrijving van de gebieden. Omdat de gebieden alleen worden gewijzigd als deze met nieuw gevonden gebieden in een volgend beeldpaar kunnen worden samengevoegd, kan temporele consistentie van het segmentatieresultaat in beeldreeksen worden verzekerd.

- *Beeldsynthese* wordt uitgevoerd met behulp van de in een gebiedsgeheugen opgeslagen parameters. Beeldsynthese is in feite een rechttoe, rechtaan proces dat betrekking heeft op de reconstructie van de gebieden vanuit de verzonden parameters en een juiste plaatsing hiervan in het beeld. Hierbij wordt bewegingscompensatie toegepast voor de linker beeldreeks en dispariteitscompensatie voor de rechter beelden. Evaluatie toont aan dat de beeldsynthese een gesynthetiseerd beeld oplevert van een hoge visuele kwaliteit. Het gebiedsgeheugen creëert de mogelijkheid voor het opbouwen van een database met daarin alle gebieden die zich voordoen in de beeldreeks. Hiermee kan de vereiste bitsnelheid verder verlaagd worden, bijvoorbeeld wanneer een eerder zichtbaar gebied achter een ander gebied is komen te liggen en vervolgens weer zichtbaar wordt.

Omdat de belangrijkste eis aan de coderingstechniek het verlagen is van het aantal te verzenden bits, moeten alle parameters efficiënt worden gecodeerd. Afhankelijk van de aard van de parameters worden verschillende *codeerstrategieën* gebruikt. Voor de parameters die beweging, dispariteit en vorm beschrijven, wordt gebruik gemaakt van verliesvrije spatiele en temporele predictiemethoden. De kleurparameters worden met behulp van de z.g. vormadaptieve DCT gecodeerd. Zodoende kunnen willekeurig gevormde gebieden efficient worden beschreven.

In het gesynthetiseerde beeld kunnen synthesefouten ontstaan als gevolg van onvolkomendheden in het bronmodel, bij occlusie, en als bij object-bewegingen de achtegrond vrij komt. Ondanks deze mogelijke synthesefouten toont een *informele subjectieve evaluatie* aan dat de visuele kwaliteit van een individueel stereoscopisch beeldenpaar vergelijkbaar is met dat van een MPEG2-gecodeerd videosignaal, ook al vereist dit MPEG2 videosignaal een aanzienlijk

hogere bitsnelheid dan de in dit proefschrift ontwikkelde gebiedsgeoriënteerde coderingstechniek.

Als de kwaliteit van de gehele beeldreeks wordt beschouwd zonder toevoeging van de synthesefouten, kunnen verschillende inconsistenties in het temporele gedrag van de gebiedsgeoriënteerde coderingstechniek worden herkend. Deze inconsistenties leveren waarneembare artefacten op die na toevoeging van de synthesefouten kunnen worden gecorrigeerd. Deze artefacten kunnen worden veroorzaakt door onjuist samenvoegen van gebieden gedurende de beeldanalyse, door temporele schokken van de gebieden als gevolg van onjuiste bewegingsvectoren, en door een plotselinge verandering van de gebiedskleur als gevolg van een ontbrekende "update" van de gebiedsparameters op een bepaald moment. Voor het verhogen van de kwaliteit van de gecodeerde stereoscopische beeldreeks wordt daarom momenteel nog de (gecodeerde) synthesefout aan de gesynthetiseerde beelden toegevoegd. Dit zal echter niet langer nodig zijn wanneer de gebiedsparameters regelmatig worden geactualiseerd.

Zelfs bij rudimentaire codering van de synthesefout - eenvoudige vectorkwantisatie als de fout boven een drempel ligt - bereikt de gebiedsgeoriënteerde coderingstechniek een vergelijkbare subjectieve kwaliteit als bij gescheiden MPEG2-compressie van de twee stereoscopische kanalen op een gelijke totale bitsnelheid.

Het onderzoek in dit proefschrift heeft alleen betrekking gehad op een model dat vaste twee-dimensionale gebieden veronderstelt die translatorische beweging kunnen ondergaan. Het blijft vooralsnog een open vraag of de efficiëntie van de parametercodering kan worden verhoogd als men gebruik maakt van een bronmodel dat uitgaat van flexibele twee- of drie-dimensionale gebieden. De coderingsefficiëntie van de synthesefout kan met behulp van een meer geavanceerde methode nog worden verhoogd, bijvoorbeeld door hierbij de gevonden gebiedsvormen te betrekken. In de naaste toekomst dient ook aandacht besteed te worden aan diverse algorithmische vraagstukken en mogelijke toepassingen van gebiedsgeoriënteerde stereoscopische coderingsmethoden. Het Europese project PANORAMA (PAckage for New OpeRational Autostereoscopic Multiview systems and Applications) is een belangrijk platform voor dit onderzoek.

Acknowledgements

This thesis was written while I was working at Siemens Corporate Technology, Department ZT IK 2 Networks and Videocommunication in Munich. Most of the work was performed under contracts arising from the European RACE II project DISTIMA.

I would like to thank my supervisor and co-supervisor, Jan Biemond and Inald Lagendijk, for accepting this work as a PhD thesis and for all the fruitful discussions which greatly influenced the content and the way the thesis was presented.

As I was working for Siemens while I was writing this thesis I would like to thank Eckart Hundt, the head of the videocommunication group. Without his support and encouragement this thesis would never have been finished or even have been started. My sincere thanks also to all my colleagues and former colleagues in the videocommunication group and to the numerous students who contributed to the success of this thesis with their stimulating discussions and programming skills. I especially want to thank Stathis Panis, Ewald Frensch, André Kaup and Thomas Riegel for their energetic support.

As a member of the DISTIMA project team, I also want to thank all the DISTIMA members for all the discussions and the pleasant atmosphere, which made work on stereoscopic video so enjoyable. Regarding this thesis, I especially want to thank Francoise Chassaing from CCETT in Rennes, Lutz Falkenhagen from the University of Hanover, Ruggero Franich from the Technical University of Delft (he is now with AEA Technology in the UK), Roel ter Horst from KPN Research in The Hague (he is now with PTT Telecom in Hilversum), Ronald Mies from KPN Research in The Hague, Kathrin Rümmler from the Heinrich Hertz Institut in Berlin and Dimitris Tzovaras from the University of Thessaloniki. They all provided a lot of information about their work in the field of stereoscopic video when I asked for their support for my thesis.

Finally, I thank my wife Ina for all her support, encouragement and patience when proof-reading my drafts and discussing the thesis in the evenings. A couple of visits to the beer garden had to be cancelled. I promise - no more PhD theses.

Stellingen

behorende bij het proefschrift

Region-Based Analysis and Coding of Stereoscopic Video

door M. Ziegler

13 oktober 1997

1. Without depth information object-based analysis-synthesis coding will never outperform existing compression standards.
(This thesis chapter 2)

   Zonder diepte-informatie zal 'analyse-synthese' codering nooit beter presteren dan bestaande compressiemethoden.
(Dit proefschrift, Hoofdstuk 2)

2. Region-based coding relies on computer graphics methods rather than on waveform coding concepts.
(This thesis chapter 4)

   Regio-gebaseerde codering steunt vooral op 'computer graphics' methoden en minder op 'waveform' coderingsconcepten.
(Dit proefschrift, Hoofdstuk 4)

3. Region-based coders are inherently variable bit rate coders.
(This thesis chapter 5)

   Regio-gebaseerde coderingsmethoden produceren onvermijdelijk een variabele bitsnelheid.
(Dit proefschrift, Hoofdstuk 5)

4. Stereoscopic video is essential for future telepresence systems.

   Stereoscopische video is essentieel voor toekomstige tele-aanwezigheidssystemen.

5. Mass acceptance of stereoscopic video demands affordable autostereoscopic displays.

   Grootschalige acceptatie van stereoscopische video vereist betaalbare autostereoscopische weergavesystemen.

6. National and international political support is essential to technological progress.

   Nationale en internationale politieke steun is essentieel voor technologische vooruitgang.

7. It is not profitable to act as prime contractor in European projects funded by the European commission.

   Het is niet lonend om als 'prime contractor' op te treden in door de Europese commissie gesubsidieerde projecten.

8. A common Europe requires conformity in the procedures and rules for getting academic degrees.

   Een gezamenlijk Europa vraagt om eenduidige procedures en regels voor het verkrijgen van een academische graad.

9. Even the uncertain perspective of curing currently fatal diseases is a sufficient reason for allowing genetic engineering.

   Zelfs het onzekere uitzicht op genezing van op dit moment nog dodelijke ziekten is voldoende reden om genetische manipulatie toe te staan.

10. Scuba diving tourism offers protection to the marine flora and fauna rather than endangering it.

    De zeeflora en -fauna wordt door de duiksport eerder beschermd dan bedreigd.

11. A Dutch summary in an English thesis written by a German is as useful as a Japanese manual for the Swiss cow-bell on an Italian bicycle.

    Een Nederlandse samenvatting in een Engelstalig proefschrift geschreven door een Duitser is net zo zinvol als een Japanse handleiding voor een Zwitserse koebel op een Italiaanse fiets.