# Disparity Estimation and Intermediate View Reconstruction for Novel Applications in Stereoscopic Video

*Anthony Mancini*

Department of Electrical Engineering
McGill University
Montreal, Canada

February 1998

---

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering.

## Abstract

This thesis addresses the problem of computing 2-D disparity fields from stereo image pairs and applying them to intermediate view reconstruction (IVR) via disparity-compensated interpolation. Intermediate views are calculated using a linear filter with angle-dependent coefficients. Two existing disparity estimation algorithms are adapted to perform IVR, and results are used for parallax adjustment of still stereo images. In one case, the well-known block matching (BM) technique is used, and several novel algorithm enhancements are proposed. The final BM scheme employs three-component estimation, a spatial smoothness constraint, and a quadtree structure. A procedure for targeting problematic blocks that requires splitting based on robust estimation is proposed, and an efficient approach for the reestimation of sub-blocks is developed. A technique for eliminating component-mismatches in stereo pairs is also examined. The accuracy of estimations based on these "balanced" images is seen to increase.

In the other case, the ill-posed problem of obtaining dense disparity maps is addressed, and the method of regularization is used to compute pixel-based vector fields for intermediate views. The conclusion is that although single image reconstruction results are comparable in both cases, the approach based on regularization is superior to the block-based scheme for a dynamic sequence of such reconstructions. Both approaches are applied to stereo parallax adjustment for still images, and numerous experimental results are included.

## Sommaire

Ce mémoire étudie le problème de calcul de champ de disparité à partir de paires d'images stéréo et de leur application pour la reconstruction de vues intermédiaires via une interpolation compensée par la disparité. Les vues intermédiaires sont calculées en utilisant un filtre linéaire dont les coefficients dépendent de la position de la vue. Deux algorithmes d'estimation de disparité de type prédictif sont utilisés pour l'ajustement de parallaxe d'images stéréo fixes. Dans le premier cas, la méthode bien connue d'appariement de blocs est utilisée et de nombreuses améliorations originales à cet algorithme sont proposées. La méthode d'appariement de blocs finale utilise une estimation sur trois composantes, une contrainte de lissage spatiale et une structure "quadtree". Une procédure pour cibler les blocs causant des problèmes et impliquant une décomposition des blocs basée sur une estimation robuste est proposée. De plus, une approche efficace pour la réestimation des sous-blocs est développée. Une technique pour éliminer les inconsistences entre les composantes dans les paires d'images stéréo est aussi développée et les estimés obtenus avec cette technique contribuent à améliorer la qualité.

Dans le second cas, le problème mal conditionné d'obtention des champs de disparité denses est étudié et la méthode de régularisation est utilisée pour calculer des champ de vecteurs basés pixels pour les vues intermédiaires. La conclusion est que même si les images fixes reconstruites sont comparables pour les deux cas, l'approche basée sur la régularisation est supérieure à celle basée bloc pour les images en mouvement. Les deux approches sont appliquées au problème de l'ajustement de parallaxe pour les images fixes et de nombreux résultats expérimentaux sont présentés.

## Acknowledgments

First and foremost, I would sincerely like to thank Dr. Janusz Konrad for extending an invitation to supervise this project and introducing me to the field of digital video processing. His innovative ideas and devotion to the project were an inspiration. It was a real pleasure to receive guidance from such a motivated, knowledgeable and insightful person. I would also like to thank Dr. Konrad for the financial assistance he has provided throughout my degree.

I would like to thank the members of the Visual Communications (VisCom) department at INRS-Télécommunications. Namely, researcher Abdol-Reza Mansouri, for his always novel approaches, constant availability for consulting and help with analysis of data. I would also like to thank him for first introducing me to Dr. Konrad! Other members of the group have also been involved in providing excellent team-work throughout, namely Dr. Stéphane Coulombe who has also helped with the translation of the abstract into french.

Of course, I wish to express my appreciation to INRS-Télécommunications in Montreál for providing me with an excellent work- and study-environment for the entire duration of my degree. Special thanks also to Mr. Anibal Jodorcovsky for his invaluable proofreading, and for pointing out more than just a few typos. This was very helpful and greatly appreciated.

In addition, I would like to thank the provincial body, *Fonds pour la formation de chercheurs et l'aide à la recherche* (FCAR), for the funding granted to me from the summer of 1996 to completion of my degree. We would also like to thank Dr. Bruno Choquet of the CCETT, Rennes, France and the RACE DISTIMA project of the European Community, as well as the NHK of Japan, for providing us with the stereoscopic image sequences used in this work.

In closing, to my family and to Silvia, thanks for all the love, support and patience; you've been a big help!

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Recently, there has been a lot of talk in the television broadcast industry about high definition TV. Interest in HDTV stems from the fact that its technology permits much more realistic and natural-looking representations of scenes as compared to existing television standards such as NTSC in North America. At a screen resolution of 1920x1036 pixels, the crispness of the image is impressive indeed. However, although there is no doubt HDTV has succeeded in largely increasing the realism of television, as is, it still lacks one very important feature; the representation of natural depth sensation. Today, it is safe to say that the next evolutionary step towards the ongoing search of increasing realism in video applications (multimedia, video conferencing, etc...) is to incorporate 3-D depth perception into the viewing experience [1, 2].

3-D video, as it is referred to, provides the viewer with the "extra" information needed for realistic depth perception to occur. *Stereoscopic* and *multiview* video form a particular sub-class of 3-D video, where the extra information provided is another, slightly displaced view of the same scene. Consider that our eyes view the world from two slightly different angles. Our brain uses differences in the two acquired projections to perceive the depth of a scene. In the same way, a stereoscopic video system acquires two views from cameras which are slightly displaced horizontally, much like the relative location of our eyes. Each acquired view is then projected to the corresponding eye, and it is through the combination of data from both views that our brain perceives depth. In multiview video, stereo scenes are captured from several viewpoints, offering a larger viewing angle to the viewer. Only two images

are presented at any time since the user selects the stereo image pair which offers the desired perspective.

The other sub-class of 3-D video is formed by techniques that "fool" the human visual system (HVS) into perceiving depth. These types of systems do not provide any additional information such as separate views for each eye, but rather use texture, shading and perspective geometry to enhance a single image and give the 3-D sensation. Such systems typically deal with the synthetic representation of objects rather than real world scenes, and are popular in the field of computer vision.

Applications for 3-D video are numerous. *Virtual environment* systems certainly benefit from 3-D video since the goal is to give the user the impression of being somewhere else. Here, depth perception is important since the environment must seem real. *Telepresence* systems also employ 3-D video by projecting the human sensory apparatus into a remote location. For this, a stereoscopic camera is placed in the remote location, and a stereoscopic display system at the local site. An example of telepresence is *teleoperation*, where the user can remotely operate a robot based on viewed stereoscopic data. Applications for 3-D video are also found in the domain of *medical imaging*, where depth perception is often essential; in particular, in 3-D laparoscopy[1] and 3-D microscopy.

Since more than one view of a scene must be captured, the stereoscopic camera consists of two lenses slightly displaced from one another. This makes stereoscopic cameras rather bulky, and impractical to move around. Stereoscopic film makers are sensitive to this, and need to be more selective of the scenes they capture. In addition, the amount of data is doubled as compared to a regular video camera with only one lens. In the case of a TV broadcast system, this doubling of transmitted information places heavy demands on the bandwidth of the information medium. Stereoscopic video compression techniques which exploit the correlation between the two perspective views are used to reduce the burden. Methods based on *disparity-compensated prediction* are often used for this. Disparity is defined as the difference in positions of *homologous* points in the left and right images of a stereo pair, i.e., points resulting from the projection of the same 3-D point onto the two image planes.

[1]Laparoscopy is direct visualization of the peritoneal cavity, ovaries, outside of the tubes and uterus by using a laparoscope. The laparoscope is an instrument somewhat like a miniature telescope with a fiber optic system which brings light into the abdomen.

*Disparity estimation* is the process of estimating the disparity for each token (e.g., pixel) in one image with respect to the other; thus, disparity is a vector. A *disparity field* is a set of vectors which, together, provide a mapping between images.

When acquiring a stereoscopic image, left and right cameras are fixed in space. Therefore together, the left and right views depict a 3-D scene from a particular viewing angle. If the stereoscopic pair is not viewed from the intended angle, an unnatural representation of the scene results. The distance between the two stereoscopic lenses is also fixed, and hence it is not guaranteed to suit the viewing characteristics of every viewer. Discomfort in viewing often results because of this. The practical problems with current stereoscopic video systems which are due to the *fixed* relative positions of cameras need to be tackled before acceptance of the technology is realized for TV broadcasting.

In the entertainment industry, stereoscopic video is used for providing 3-D movies to the public. The IMAX® Corporation of Ontario, for example, produces high-quality stereoscopic movies on 8-story-high screens. The screens are large enough to cover the viewer's entire field of view. In this case, viewer head movements are negligible as compared to the size of the screen, so the distortions resulting from incorrect viewing angle discussed above are avoided. However, in the case of broadcast TV or computer monitors where much smaller screen sizes are used, viewer head movements pose an important problem.

Today, this problem is solved digitally through the reconstruction of *intermediate*, or *virtual* views. Intermediate views permit the display of the same scene, but from a different viewing angle. Hence, as the viewing angle changes, appropriate intermediate views are computed and displayed, and the distortions that normally result are avoided. The so-called problem of intermediate view reconstruction (IVR) therefore offers *continuous look-around* to the viewer as the viewing angle is changed. *Parallax adjustment* can be made possible since intermediate views permit adjustment of the distance between cameras to suit a particular viewer's preference. The reconstruction of these virtual views can also be used for the application of *missing frame replacement*. Here, known data in a sequence of images is used to interpolate a missing frame.

Based on the above, we believe 3-D video to be a next-generation medium that

will revolutionize information systems. The creation of the 3-D image communication and broadcasting systems will require the development of various technologies. Among them will no doubt be the technology of intermediate view reconstruction for stereoscopic video. This thesis offers novel solutions to the IVR problem.

As mentioned, disparity estimation is used to remove redundancies in stereoscopic video compression systems. The approaches towards IVR presented in this paper are also based on the process of disparity estimation, but performed as a function of the desired intermediate view position. The resultant disparity field is then used to reconstruct the intermediate view at that position. To date, various approaches to disparity estimation have been proposed. In this thesis, two existing approaches have been adapted to the problem of IVR. In the interest of high quality view reconstruction, various improvements have been proposed to cure existing problems.

## 1.1 Problem statement

This thesis address the problem of reconstructing virtual intermediate views by first solving the correspondence problem using disparity estimation. Using the estimated vector field, reconstruction is performed using simple two-coefficient linear interpolation. The main focus of the thesis is in obtaining accurate disparity vector fields in order to produce high quality image reconstructions. Envisaged applications are in the area of entertainment, where small baseline distances are used between (almost) parallel left and right video cameras, and where arbitrary natural scenes are typically acquired.

## 1.2 Organization of the thesis

The layout of this thesis is as follows. The following chapter will provide the reader with the necessary background and fundamental concepts relating to the field of stereoscopic video. The geometric principles involved are considered, and the different aspects of a stereoscopic video system, from acquisition to display, are presented.

Chapter 3 gives an overview of past work done in the field of intermediate view reconstruction. Existing techniques found in the literature are presented. Past work done in the field of disparity estimation is also presented, and a few algorithms are

discussed in the context of disparity-compensated prediction. The tasks of IVR and disparity estimation are defined in detail.

Chapter 4 discusses the adoption of a model on which the proposed algorithms for IVR are based. The model places a simple constraint on the process of disparity estimation so that the resultant vector field can be used to reconstruct a virtual view. Two existing techniques for disparity estimation are adapted to perform IVR, and various improvements are proposed. One technique is a block-based estimation, and the other, pixel-based. Both techniques are adapted to the model. The presented algorithms are implemented in software, and resulting disparity fields are shown throughout the thesis. The goal is to obtain accurate disparity fields since this has a direct impact on the quality of reconstructed views. A technique for reducing the execution time of the disparity estimation algorithms is proposed, and a well-known robust interpolator function presented.

Chapter 5 focuses on the intermediate view reconstruction results obtained based on the disparity fields estimated in Chapter 4. The quality of the reconstructed views is assessed, and a comparison between the block- and pixel-based approaches is offered.

Finally, Chapter 6 presents a summary of the contributions of this thesis, and provides suggestions for future work.

## 1.3 Contributions

This thesis proposes several novel improvements to the simple template- or block-matching estimation algorithm seen in motion estimation. First, a pre-processing stage which eliminates global component-mismatches between homologous tokens in the left and right images is implemented. The approach was proposed by the MPEG-2 Multi-View Profile group to eliminate luminance mismatches, and we extend it to remove chrominance mismatches as well. Experimental results included herein show that this preprocessing stage significantly improves the quality of reconstructions.

The simple block-based scheme is then adapted to perform estimation over all three image components in order to eliminate ambiguous matches in the image where luminance detail is low. Although this increases the complexity of the algorithm, it also improves the accuracy of the estimated disparity field.

The block-matching scheme is then adapted to perform smoothness via regularization. This algorithm enhancement offers the greatest gain in terms of quality of reconstruction as it results in a very smooth (regular) disparity field.

Finally, the approach is modified to improve reconstructions near object boundaries in the images. Local depth constancy is relaxed in these areas by using a quadtree-based splitting structure; smaller block sizes are used for estimation. A robust technique for identifying problematic blocks which cover object-overlap regions is proposed, and experimental results are shown.

# Chapter 2

# Stereoscopy, the Fundamentals

This chapter is devoted to presenting the basic concepts of stereoscopic video in the context of binocular viewing. First, how the human visual system (HVS) provides depth perception to the brain is examined. Next, how this information of depth is reproduced on a 2-D display for viewing is looked at. Finally, the stereoscopic display system itself, and potential image distortions it may cause, are examined.

## 2.1 Geometric considerations

### 2.1.1 Geometry of the human visual system

Understanding the geometry behind how our eyes perceive depth in a real world scene will lead to a better understanding of how to build stereoscopic display systems. The geometry of the HVS's binocular vision is examined here.

Consider the sketch in Figure 2.1 where the eyes' reaction to different points in a scene is illustrated. Both left and right eyes are fixated on a point p1 in space, forming the angles $\delta_1$ and $\delta_2$ as shown. This focuses an image in the center of the fovea of each eye. A point p2 elsewhere in space (at a different depth than p1) produces another image in each eye, but each may be at a different distance from the fovea. That is, $\delta_1 \neq \delta_2$, and the sum of these angles is referred to as the *retinal disparity*. As illustrated, the angle measured from the fovea towards the inside of the eye is positive. It is the retinal disparity which provides the brain with information towards the depth and shape of an object when the two images get fused together.

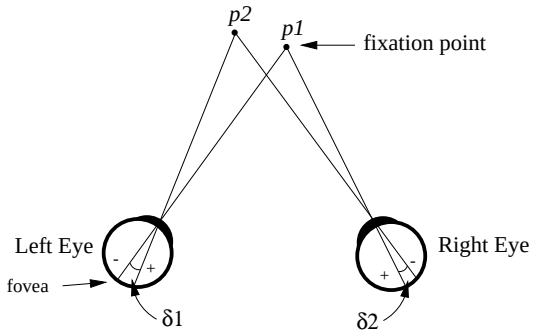



**Fig. 2.1** Visual system geometry showing retinal disparity, $\delta_1+\delta_2$, based on the fixation point $p1$.

To further demonstrate, consider the sketch in Figure 2.2. As the two eyes focus on the fixation point $p1$, they form the convergence angle, $\alpha$. Similarly, the convergence angle of point $p2$ is $\beta$.

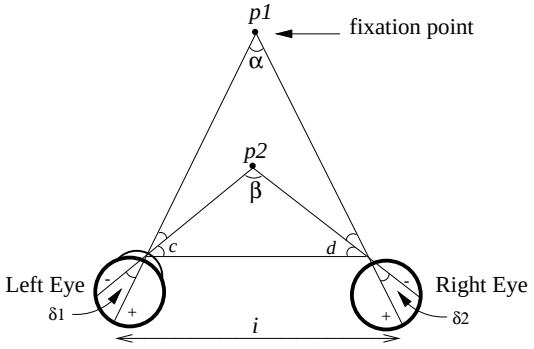



**Fig. 2.2** Visual system geometry showing convergence angles, based on the fixation point $p1$.

Clearly, $\alpha + \delta_1 + c + \delta_2 + d = 180$, and $\beta + c + d = 180$, so that

$$\alpha - \beta = \delta_1 + \delta_2. \tag{2.1}$$

This means that the difference in the angles of convergence defined by two points in the real world scene is given by the retinal disparity. The depth perceived depends on whether the point $p2$ is closer or farther than the fixation point. If it is closer, then $\beta > \alpha$, and the retinal disparity will be negative. This corresponds to what is called

*crossed disparity* since the eyes must cross to focus on the point $p2$. Otherwise, the retinal disparity is positive, referred to as *uncrossed disparity*[3].

The important conclusion to be drawn here is that the depth our brain perceives is related to retinal disparity. In fact, according to Hodges & Davis, they are related in a monotonic, non-linear manner [3]. In general, for a fixed distance for $p1$, a larger depth between $p1$ and $p2$ corresponds to a larger difference in convergence angles, hence a larger magnitude for the retinal disparity.

### 2.1.2 Epipolar geometry

The fact that a stereoscopic video system is made up of two cameras observing the same real world scene imposes some constraints on the two resulting images. In order to reconstruct 3D coordinates from a pair of given 2D images, one must first deal with the *correspondence problem*: given a token in the left image, what is the corresponding token in the right image? Since there are too many potential pairs of tokens which correspond, some properties must be exploited in order to come up with one solution. The fundamental constraint typically used is the the epipolar geometry constraint.

I like the figure from Naemura, Kaneko, and Harashima [4] depicting the idea behind epipolar geometry. It is repeated here as Figure 2.3.

The *epipolar plane* is defined by a point $P$ in space and the line joining the two lens centers with focal length, $f$. *Homologous epipolar lines* are defined as the intersection of the epipolar plane and the image planes, as shown in Figure 2.3. All points from a particular epipolar plane are projected onto a corresponding pair of homologous epipolar lines of the image planes.

Due to this geometrical constraint, any point lying on an epipolar line in one image corresponds, necessarily, to a point lying on the homologous epipolar line in the other image. This fact is the basis of all stereo matching methods.

One typical assumption that is made to help deal with the very complex correspondence problem is that parallel cameras are used to acquire the stereoscopic image, and that the cameras are aligned (calibrated) so that there is no vertical shift between them[1]. Section 2.2.2 talks about this in more detail, but it is important here

[1]Camera calibration implies that all extrinsic camera parameters are known, such as position, angle, etc...

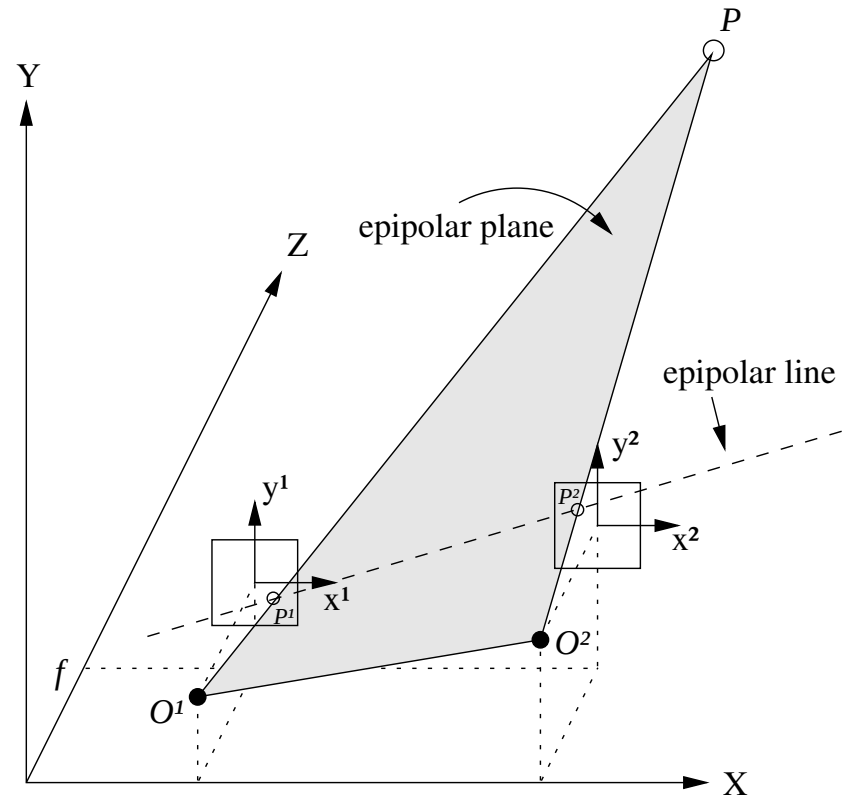



**Fig. 2.3** Camera imaging geometry modeled as a perspective projection showing the epipolar constraint. $O^1$ & $O^2$ are the camera optical centers which form their respective image planes a distance $f$ away.

to mention that the parallel camera setup benefits from parallel epipolar lines. This is important because it implies that in this case, the stereo images do not have any vertical parallax between them, and hence the task of finding correspondences between them becomes a 1-D problem (i.e., only horizontal disparity vectors are assumed to exist).

There are two major elements of our work. The first is solving the correspondence problem (disparity estimation), and the second is intermediate view reconstruction. In the context of disparity estimation, the parallel camera assumption *was not* made since small vertical screen parallax was permitted. However, as will be discussed in greater detail in Chapter 5, this assumption *was* made in the context of intermediate view reconstruction.

### 2.1.3 Viewing geometry

The geometry behind what each eye sees in a 3-D scene has been studied. The geometry of modeling stereoscopic displays to achieve the same effect of depth when viewed on a two-dimensional plane is now examined. In particular, the concepts of *crossed* and *uncrossed* disparity can be discussed in the context of a geometric model by introducing the notion of *screen parallax*.

By tracing the projection vector of a point $P$ in 3-D space to its location on a two-dimensional display, it is clear that the projection on the left eye of this point is quite different from that on the right eye. The distance between the points $P_{left}$ and $P_{right}$, shown in Figure 2.4, is defined as the screen parallax, $p$ [3].

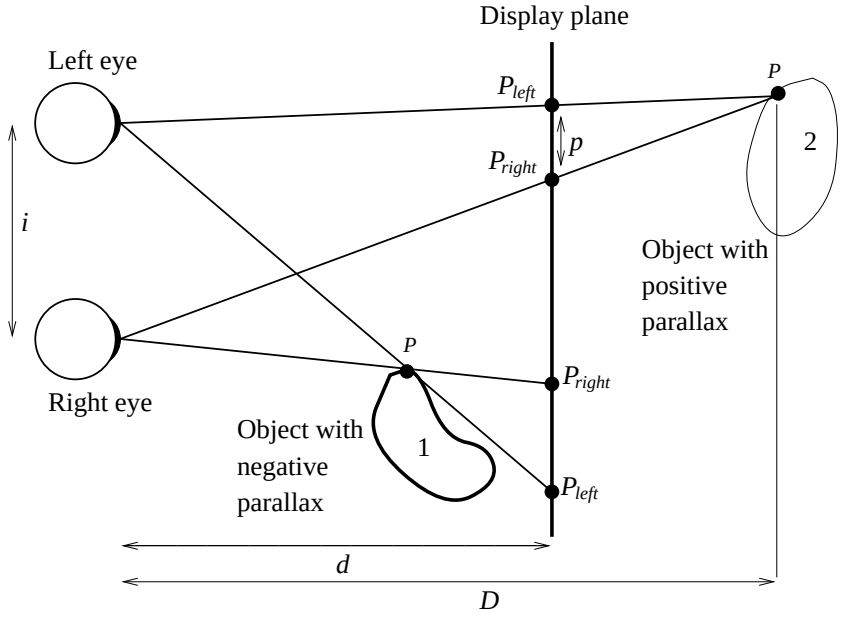



**Fig. 2.4** Top view of a stereoscopic geometric model showing screen parallax induced by left- and right-eye projections from the point $P$ in 3-D space.

*Negative parallax* results when $P_{right}$ is to the left of $P_{left}$ on the display. In this case, the model reproduces the effect of crossed retinal disparity defined in Section 2.1.1. To associate negative parallax with crossed disparity, consider the display plane as being the fixation point. Then, as in the case of crossed disparity, the eyes

would have to cross to focus on point $P$, which gives the impression of being closer to the viewer than the fixation point.

*Positive parallax* occurs when $P_{right}$ is to the right of $P_{left}$ on the display, and this effect can be compared to uncrossed retinal disparity. Naturally, the screen parallax is equal to zero for any object perceived to be at a distance $d$ from the viewer.

Screen parallax and retinal disparity are not *directly* related, however. On the one hand, retinal disparity is measured on the two retinae and its value is dependent on the convergence angle and the viewer's focus point. Screen parallax, on the other hand, is measured directly from the display, and its value depends on how points in the real world scene are mapped to the display, as well as on the viewing distance. Conceptually however, screen parallax induces retinal disparity which in turn provides the stereoscopic cue needed for depth perception.

The model shown in Figure 2.4 demonstrates how stereoscopic display systems "trick" the brain into using its binocular vision. The idea is that the model produces two sets of points, each set making up an independent representation of the real world scene (i.e., an image). Given these two images now, there is a need to separate them so that the viewer's left eye sees only the "left" image, while the right eye sees only the "right" image. How this is done using special glasses and high-frequency display monitors in the context of a stereoscopic video system is discussed in more detail in Section 2.2.3.

Looking at object 2 of Figure 2.4, it is easy to show that the amount of parallax on the display required to achieve a given depth perception, $(D - d)$, is given by:

$$p = \frac{i(D-d)}{D}, \tag{2.2}$$

where $D$ is the distance from the point to the viewing plane, $d$ is the distance from the display to the viewing plane, $i$ is the human inter-ocular (pupillary) distance, and $p$ is the screen parallax.

In general, as the object moves farther away from the display plane in either direction, the magnitude of the screen parallax increases, and hence the distance between left and right eye image projections on the display also increases. Similarly, it was seen that the farther two points were from each other in a real world scene (in

the $z$-direction), the greater the magnitude of the retinal disparity. This confirms the tight relationship between retinal disparity and screen parallax.

## 2.2 Stereoscopic video system

The display process of a stereoscopic video system is made up of three separate coordinate transformations. From the real world object space, the cameras' CCD imaging sensors transform this real-world data into two separate sets of two-dimensional coordinates. These are then transformed to the coordinates of the physical display monitor. Lastly, our eyes transform the displayed image to the final image space, our brain. The procedure is summarized by the following three-stage process:

| Object space | $\Rightarrow$ | CCD coordinates | $\Rightarrow$ | Screen coordinates | $\Rightarrow$ | Image space |
|---|---|---|---|---|---|---|
| (3-D) | | (2-D times 2) | | (2-D times 2) | | (3-D) |

### 2.2.1 Image acquisition

The first stage of coordinate transformations shown above is accomplished with two cameras placed side by side acquiring the real world data. The *interaxial separation* of the cameras, or the distance between lenses used to take a stereoscopic photograph, has a large impact on the strength of the stereoscopic cue. Typically, a separation distance equal to the average adult pupillary distance, or 64 millimeters, is used. Since actual pupillary distances vary from individual to individual, the strength of the stereoscopic cue, or degree of "3D-ness", that the viewer experiences, will vary. The closer the lenses are together, the more the screen parallax of objects in the scene is reduced and thus the more the stereoscopic depth effect is reduced.

It is interesting that some people have trouble fusing certain stereoscopic images. One reason for this could be that the particular interaxial separation used to take the photograph is not well matched to the particular viewer's inter-ocular distance. The diagram in Figure 2.5 demonstrates what happens when the screen parallax is greater than the pupillary distance (i.e., when the interaxial separation is too large, resulting in a very strong stereoscopic depth cue).

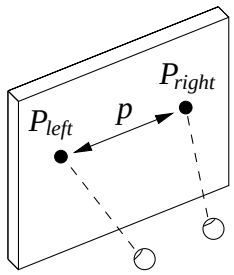



**Fig. 2.5** Visual system geometric model showing exaggerated stereoscopic depth cue and diverging eyes ("hyper stereo").

In this case, the eyes must diverge in order to fuse together the stereoscopic image, something that does not occur when looking at objects in the visual world. This situation, referred to as *divergent parallax*, often results in discomfort to the viewer due to the unusual strain it places on the muscles of the eye. It would be beneficial if the viewer could avoid this discomfort by adjusting the interaxial separation of the cameras, hence the screen parallax, to fit his/her particular pupillary distance. Of course, the stereoscopic image can only be acquired using a fixed interaxial separation, so how to adjust this parameter? In fact, this idea is one of the major motivations of the project behind this thesis work, and will be discussed in much greater detail in the upcoming chapters.

The amount of "permissible" screen parallax[2] is a function of the viewing distance. The further a viewer is from the display screen, the larger is the permissible amount of screen parallax. From the usual workstation viewing distance of about 45 cm, the general rule is not to exceed positive or negative screen parallax values of about 12 millimeters [5].

Screen parallax can also be expressed in terms of the viewing angle [5]. Consider Figure 2.6; given an amount of screen parallax $p$ and a viewing distance $d$, the viewing angle $\beta$ can be described by:

$$\beta = 2 \arctan \frac{p}{2d}. \tag{2.3}$$

[2]The term "permissible screen parallax" is meant as the largest amount of screen parallax which does not cause viewer discomfort for a typical adult viewer.

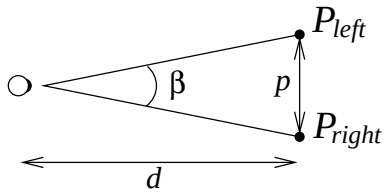



**Fig. 2.6** Relationship between screen parallax $p$, and the viewing angle $\beta$, given by (2.3).

### 2.2.2 Pick up equipment

A stereoscopic video acquisition system consists of two (or more) cameras positioned side by side to obtain left and right images. Both cameras should have the same focal length values, $f$. In fact, the cameras should be calibrated so that there is little or no imbalance between them in any way. Even small imbalances in the cameras' focal lengths, for example, could seriously bias solutions of the correspondence problem. This in turn will have a negative impact on the quality of reconstructed images.

There are three common configurations of the video camera pair. The simplest of three uses two cameras with parallel lens optical axes and parallel camera optical axes. In this scenario, the common field of view between the left and right acquired images becomes very small (depending on the interaxial separation, of course), and the correspondence problem becomes impossible for tokens at the outer extremes of the images.

The second configuration, the *toed-in* approach, has cameras which are rotated towards each other so that both their camera and lens optical axes coincide, and converge at a so-called *convergence point*. This increases dramatically the common field of view between the cameras, but suffers from complexity of setup.

The third configuration places cameras so that their lens optical axes are parallel to each other, but their camera optical axes converge at the convergence point. This is achieved by horizontally shifting the CCD sensors in the cameras to obtain the shifted images on the display [6]. This configuration is called the *parallel* configuration, or

*convergence by lateral shifting*, and is simpler to set up than the toed-in configuration. It also benefits from a large common field of view. The two are illustrated in Figure 2.7.

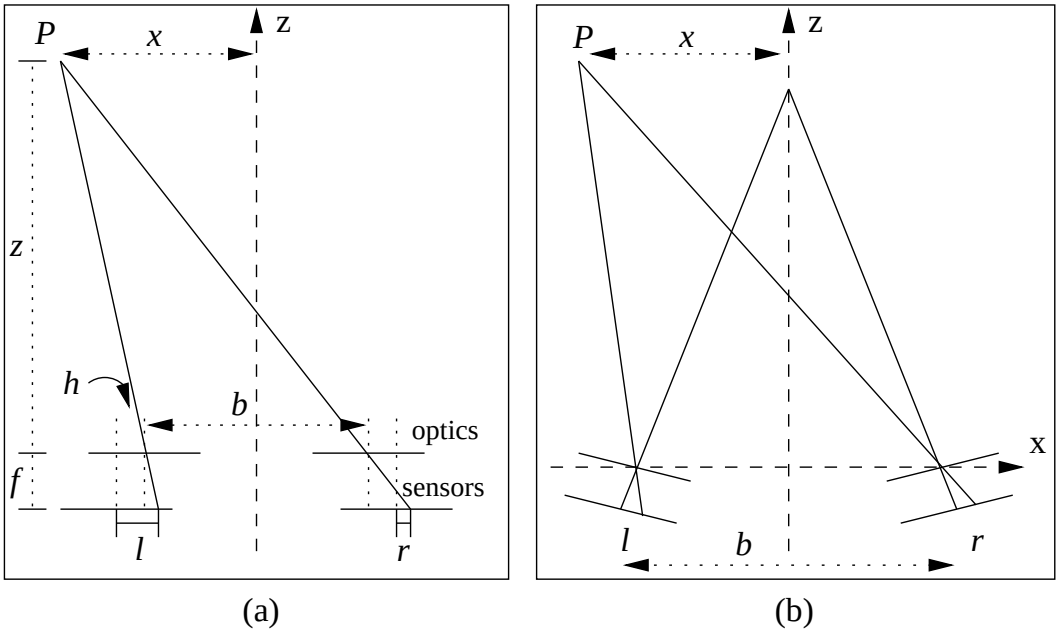



**Fig. 2.7** (a) Parallel and (b) toed-in stereoscopic camera configurations.

The parallel camera setup is advantageous since there is no vertical disparity introduced, and since the governing disparity equation (i.e., the difference of the abscissas $l$ and $r$ of the images of point $P$ on the sensors) is straightforward [7]. From simple trigonometric principles applied to Figure 2.7 (a), the following relation is found to hold:

$$\frac{x - \frac{b}{2}}{z} = \frac{x - \frac{b}{2} - h + l}{f + z}. \tag{2.4}$$

Solving for $l$, we get

$$l = \frac{fx}{z} - \frac{fb}{2z} + h. \tag{2.5}$$

Similarly, solving for $r$,

$$r = \frac{fx}{z} + \frac{fb}{2z} - h. \tag{2.6}$$

To calculate the disparity, we get

$$d_0 = (l - r) = 2h - \frac{fb}{z}, \tag{2.7}$$

which is a simple relationship describing the disparity in terms of the focal length $f$, the baseline distance $b$, the object distance $z$, and the CCD sensor shift $h$. Hence we see that for parallel cameras, the disparity is inversely proportional to depth. Also, this expression is independent of point $P$'s lateral position, $x$.

The toed-in configuration is simpler to set up and often used in practice because it maximizes the common field of view between the cameras. However, it is more complicated to analyze mathematically. The governing horizontal *and vertical* disparity equations are not simple, and depend on the lateral position, $x$. The reader is referred to [7] for more details on the mathematical relationships describing the toed-in camera configuration.

In summary, the parallel setup benefits from simpler mathematical expressions and also simplifies the correspondence problem, yet it is more impractical to set up. On the other hand, the vergent (toed-in) setup suffers from vertical parallax. As we have seen, vertical parallax is undesirable since it complicates the already complex correspondence problem[3]. To exploit the favourable characteristics of the parallel setup while still allowing a vergent camera setup, *rectification* can be used to remove, to a large extent, the vertical parallax from images acquired using the toed-in setup. This technique is a preprocessing stage, and the reader is referred to [8] for implementation details. All correspondence experiments conducted within the framework of this project were done on images acquired from cameras which were slightly toed-in. However, the parallel camera assumption is not made, and hence image rectification not needed.

[3]It has also been verified in [6] that only a small degree of vertical parallax is tolerated before discomfort is felt by viewers.

### 2.2.3 Display technologies

#### Stereoscope

Probably the very first attempt at stereoscopic viewing was the *stereoscope*, invented by Charles Wheatstone in the 1830's. The concept behind this display system was very simple. The idea was that two images (camera snap-shots) were taken of a scene, with the cameras slightly displaced from each other horizontally. The easiest way to describe how this mechanism works is by looking at the somewhat crude drawing shown in Figure 2.8.

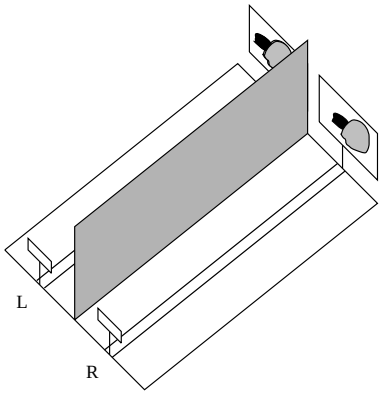



**Fig. 2.8** First generation stereoscope.

The stereoscope physically separates the left and right perspective views using a vertical separator. This completely eliminates the possibility of crosstalk between the views. At the time, the stereoscopic pictures were black and white and of a poor quality. However, the interesting thing is even then, human binocular vision was well understood since the stereoscope had an adjustable pupillary distance between the viewer lenses, as well as an adjustable viewing distance. Together, these two adjustments allowed the viewer to achieve his/her optimal stereoscopic cue settings. The stereoscopes of today are much more advanced and employ mirrors with high quality pictures to achieve the stereoscopic cue.

Due to complexity of setup (precise viewing angle, precise alignment of left & right images, etc ... ), stereoscopes are typically only used in psycho visual experiments.

### Anaglyphs

This type of stereo display system employs spectacles to separate left- and right-eye images. Left- and right-eye lenses are red and blue colour-coded, respectively, and act as visual filters. The red-blue anaglyphic process is based on the patents of Ducos du Hauron who described the system in 1858. The system requires that the left image be dyed red and the right image be dyed blue (or vice versa). The two images are then printed superimposed on the page (giving the out-of-focus look). The red and blue lenses then allow each eye to view the page separately and give the stereoscopic cue.

### Autostereoscopic

In the autostereoscopic display system, left and right views are spatially multiplexed as in the anaglyphic display, but the viewer is not required to wear any special spectacles. The separate images of a stereo pair reside exclusively on either the odd or even columns of the display. The interleaved pictures are then directed to the viewer's eyes by means, for example, of a lenticular sheet at the display surface.

### Polarization

This system uses passive spectacles, and has a large liquid crystal polarizing device attached to the display screen. Alternate left and right frames are encoded with either a clockwise or anti-clockwise polarization. The polarizing spectacles then decode the correct perspective view for each eye.

### Time-sequential displays

The stereoscopic display system used within the context of this work consists of a "stereo-ready", multi-sync monitor capable of operating at twice the typical refresh rate of 60 fields per second. To achieve the stereoscopic cue, left and right images are spatially superimposed and temporally interleaved on the monitor. The left and right images are alternately displayed at the rate of 120 fields/sec, sequenced as follows: left, right, left, right, etc ...

The viewer wears CrystalEyes active Liquid-Crystal (LC) shuttering glasses which alternately block and unblock the images in synchronization with the monitor's field rate. In this way, while the left image is being displayed on the monitor, the right shutter of the active glasses blocks the view of the right eye so that it sees nothing, and vice versa. The net effect is that the each eye sees its intended perspective view only. Furthermore, the refresh rate of 120 fields per second is sufficiently high to prevent flicker[4].

### 2.2.4 Display distortions

It has been seen that in order to properly model and implement a stereoscopic display system, a solid understanding of the geometry involved is required. Furthermore, one must be sensitive to potential distortions produced by the display system itself. This section takes a brief look at some of the various stereoscopic display distortions; those that are independent of the display technology, as well as those that are particular to the time-sequential displays used at INRS-Telecommunications.

#### Depth plane curvature

The toed-in camera configuration causes a curvature of the perceived depth planes. Hence, fixed depth planes are not perceived as flat in image space. The parallel camera configuration, however, results in flat depth planes which are parallel to the display monitor. Given a line of objects along any one depth plane, this form of distortion results in objects at the corners of the image perceived as being farther from the viewer than objects at the center of the image. It could also lead to wrongly perceived relative object distances on the display, and to disturbing image motions during panning of the camera system. This display distortion is independent of the display system.

[4]This is not always true since many of the sequences used throughout this project have a refresh rate of 50Hz, or 100Hz in stereo mode. For these sequences, the flickering is noticeable.

### Depth non-linearity

It has been stated in [6] that the distance from the cameras to the object does not correspond linearly to the distance perceived by the viewer to the stereoscopic image. Figure 2.9 demonstrates that the depth is stretched between the viewer and the monitor, and compressed between the monitor and infinity. This naturally leads to incorrect depth perceptions. In the figure, convergence and viewing distances are equal to 1m. Both camera configurations suffer from depth non-linearity. This display distortion is independent of the display system.

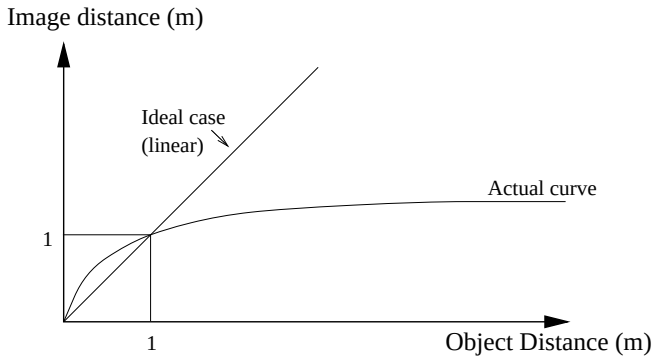



**Fig. 2.9** Image distance vs. object distance showing depth non-linearity.

It has also been stated in [6] that a linear relationship between image and object depths can be achieved if and only if the depth of an object at infinity is displayed at image infinity. However, this may not be altogether possible. Consider the following reasoning.

An experiment has been carried out to help determine what role the human visual system has to play in such effects. As we already discussed, it has been suggested that there are limits as to the amount of screen parallax humans can tolerate for stereoscopic fusion to remain possible. By increasing and decreasing the screen parallax between two images on a stereoscopic display until fusion was no longer possible, it was found that indeed some people had a greater ability to fuse the images together than others [6]. Ten subjects were studied, and it was found that some could only tolerate a small degree of parallax, while others had a much larger range of tolerable screen parallax.

The results of this experiment suggest that if the goal is to satisfy the greatest number of viewers, then the depth range should be kept to a minimum. However, this constraint directly opposes the requirements for a linear depth relationship mentioned above. Depending upon the depth range of the real world scene being captured, this condition may not be possible and depth non-linearity often becomes unavoidable.

### Shear distortion

Another image distorting phenomenon occurs with stereoscopic displays whenever the viewer changes viewing positions. In the case of a display which is inherently two-dimensional, the stereoscopic image appears to "follow" the viewer as he displaces himself from side to side. This leads to what is called *shear distortion* as the image is sheared about the surface of the monitor. In essence, objects perceived to be in front of the monitor shear in the direction of motion, while objects behind the monitor shear in the opposite direction. This display distortion is a direct result of the fact that only two perspectives of the scene are shown, which are taken from a fixed set of stereoscopic cameras.

As a solution to this problem, one could imagine a display system that could track the viewer's viewing position, and display the scene from the appropriate angle. This form of *continuous look-around* stereoscopic imagery is quite expensive, however, as multiple cameras must be used to acquire enough information from the real world scene. Typically five cameras are used to acquire a discrete set of viewing angles, and then intermediate view reconstruction techniques are used to provide missing viewing angles within a reasonable viewing range. This approach solves the shear distortion problem, but it requires a tracking device to know the viewer's viewing angle at all times from which to calculate the correct spatial image. Both camera configurations suffer from shear distortion.

### Keystone effect

This form of distortion is particular to the toed-in camera configuration, and results in a vertical parallax between the left and right images at certain areas of the display as shown in Figure 2.10.

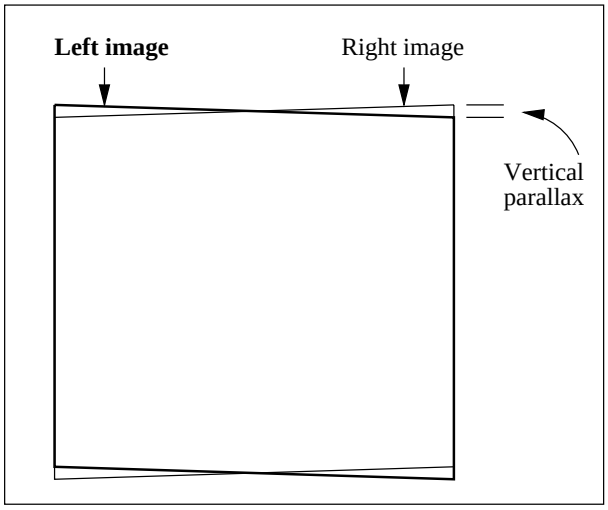



**Fig. 2.10** Induced vertical parallax due to Keystone distortion from toed-in camera configuration.

This distortion is a result of the fact that the imaging sensors of the two cameras are not in the same plane. The effect increases as the distance between the two cameras increases, and it is greatest in the corners of the display. The keystone effect also induces additional horizontal parallax in the display, and is in fact the source of depth plane curvature discussed above [6]. The parallel camera configuration does not suffer from either the keystone effect, or depth plane curvature. This display distortion is independent of the display system.

### Image imbalances

Since left and right cameras have their own set of electronic circuitry (i.e., CCDs, amplifiers, etc ... ), colour imbalances are bound to exist since it is rather difficult and costly to calibrate cameras *exactly*. Also, luminance imbalances are often present, primarily due to the different angles from which the camera "sees" the scene. This is more important for the convergent camera setup.

Apart from the fact that this distortion can affect the stereoscopic viewing experience, it can also seriously bias the correspondence problem. Some preprocessing in the form of balancing is often beneficial to the disparity estimation problem discussed in Chapter 4. A similar type of situation exists when there is an imbalance in the focus

of the left and right cameras. These types of imbalances which may cause display distortions are independent of the display system.

### Image crosstalk

It is inevitable for a stereoscopic display which interleaves two images at once, either spatially or temporally, to suffer of some degree of image crosstalk, or *ghosting*. It is impossible to completely remove the left image from the view of the right eye, and vice versa. The net effect is that certain parts of the perspective views are seen around fused objects, in an attenuated form of course.

It is interesting that crosstalk was shown to be a function of the phosphor persistence [9]. That is, the phosphors on a regular display monitor are excited by electron beams to achieve a certain luminance or colour, and it takes a finite length of time to change this luminance value. In an ideal field-sequential stereoscopic display, the image of each field, made up of glowing phosphors, would vanish completely before the next field was written. In practice, however, this is not the case. After the left image is written, it will persist while the right image is being written. The effect of ghosting is dependent on image brightness/contrast and on the amount of screen parallax. As presented, this form of display distortion is particular to the time-sequential display system.

### Motion parallax distortion

In a stereoscopic video system, left and right cameras synchronously acquire a scene. However, in the case of a time-sequential display system, left and right images are sequentially displayed on the viewing monitor. That is, images acquired at the same time are displayed $1/120s$ apart on the display monitor. This lapse of time is given by the reciprocal of the vertical refresh rate, which is typically 120Hz. Although once should be aware of this problem, it does not cause any perceptible problems for stereoscopic viewing.

### Negative parallax

Negative parallax, presented in Section 2.1.3, is not a display distortion in itself. On the contrary, it is responsible for probably the most interesting aspect of stereoscopic video: objects that are perceived to be coming out of the display monitor. However, stereoscopic film makers must be careful with negative parallax. It is important to avoid situations where objects with negative screen parallax fall at image boundaries. This results in slight discomfort to the viewer since the viewing of the physical edge of the display monitor right next to such an object takes away from the 3-D experience. In essence, it creates an inconsistency between a perceived occlusion and reality.

On the other hand, objects with positive parallax which fall next to the edge of the monitor do not cause any viewer discomfort. With this scenario, it is as if the viewer is looking through a window where the monitor's edge becomes the window pane which causes the perceived occlusion. This is perfectly natural for the viewer.

# Chapter 3

# Disparity estimation and intermediate view reconstruction – a review

This chapter examines the processes of disparity estimation and intermediate view reconstruction (IVR). First, a complete definition of IVR is presented, including a motivational discussion on the need for IVR. Some existing techniques are examined and classified according to their approach as either 3-D model-based techniques (typically used in computer vision), or 2-D signal processing techniques. This is followed by a high-level overview of the process of disparity estimation, the main ingredient needed for IVR. Finally, the chapter concludes with a look at a few existing disparity estimation techniques.

The project focuses on 3-D viewing for entertainment purposes. There will be references made throughout this chapter to the IMAX large-screen stereoscopic films, which employ a pair of cameras, "left" and "right", for image capture. The left and right images are displayed either through spatial superposition, or temporal interleaving; viewers are required to wear passive polarized spectacles with the former, and active LC shutter glasses with the latter (see Section 2.2.3 for more information on stereoscopic display technologies).

## 3.1 Intermediate view reconstruction

One major problem with traditional stereoscopic video display systems, which I have already alluded, to is the so-called *continuous look-around* problem. Best described by *shear distortion* (Section 2.2.4), the 2-D stereoscopic display suffers from being incapable of displaying anything but one particular view of a scene as captured by the original left and right cameras. This results in an unnatural representation of the scene whenever the viewer displaces himself laterally away from the center viewing position. Naturally, when dealing with eight-story high IMAX-size screens, the relative movement of viewers' heads is negligible. Even a large lateral shift of the viewing angle would not necessitate any perspective change. However, when viewing an ordinary computer monitor or television screen, even small viewer head movements could result in an unrealistic representation of the real-world scene.

In an ideal scenario, the displayed stereoscopic image would be a function of the viewing angle. All significant lateral viewer head movements would prompt a switch of the displayed image, in real-time[1]. This type of system would require a head-tracker, for example, to determine the viewing angle, and depending on how strict the system is to viewer movements, could offer *continuous motion parallax* to the viewer (i.e., no "flipping"-artifacts as the viewer displaces himself, only a continuous representation of the scene).

One other problem with stereoscopic displays, which I have again tried to motivate in earlier discussions, is the problem of *fixed* interaxial separation. Stereoscopic cameras are displaced by a distance equal to the average adult inter-pupillary distance, or about 64mm [5]. This means that the acquired stereoscopic images will be well-suited for someone with a *similar* inter-pupillary distance. However, what about viewers with a much smaller distance that would not be able to tolerate such a strong depth cue? Or, what about people with larger separations that could tolerate, and perhaps prefer, a stronger depth cue? By no means can the film maker shoot the stereoscopic scene from a multitude of camera-pair positions! Instead, in the ideal case, only one stereoscopic image is filmed, but each particular viewer would be able to adjust the camera interaxial separation, thereby affecting the amount of screen par-

[1] With systems that offer such perspective changes based on viewing angle, the amount of tolerated lateral head shift before a perspective change is required, varies.

allax and hence adjusting the stereoscopic cue. Much like the original stereoscopes of the 1830's, each viewer would be able to achieve his optimal stereoscopic viewing experience by adjusting the virtual baseline distance.

These two problems just discussed are directly related. Both are due to the fact that the original camera positions are not well-suited under certain conditions (i.e., viewing angle), and for certain people (i.e., those with inter-ocular distances significantly different than those assumed by the film maker). Both require the display of stereoscopic images acquired from positions *different* than the original camera positions. In essence, the solution is *intermediate view reconstruction*.

The task of intermediate view reconstruction consists of synthetically producing images that would be acquired from a "virtual" camera located anywhere in the vicinity of the original left and right cameras. Consider the drawing in Figure 3.1. The original left and right cameras are denoted by "L" and "R". Someone with a


**Fig. 3.1** Stereoscopic cameras showing original ("L" and "R") and virtual (1, 2, 3 and 4) camera positions.

smaller inter-ocular distance might prefer viewing stereoscopic images acquired from cameras "1" and "2". This camera pair would offer the same view of the scene, but with a reduced depth cue and with potentially greater comfort to the viewer. On the other hand, someone who would prefer a stronger depth cue would select cameras "3" and "4", which again maintain the viewing angle. These scenarios correspond to the *parallax adjustment* application, where the existence of intermediate views allows the viewer to adjust the degree of "3-D ness" in the image by adjusting the screen parallax induced by the camera separation.

Similarly, any viewer who is comfortable with the fixed camera positions, but who shifts his head laterally parallel to the display axis would expect to see a slightly different view of the scene. This viewer would opt for the "3"-"2" camera pair. This corresponds to the *continuous look around* problem, where the viewer may shift his

head with respect to the display, and still visualize a realistic representation of the scene.

In summary, to allow for parallax adjustment and continuous look around applications, there is a need to reconstruct images as acquired from a virtual camera. This virtual camera could be located at positions "1", "2", "3", "4", or anywhere in the vicinity of cameras "L" and "R". The images acquired from the original positions are the data used to synthesize images as seen from some other arbitrary position. In the context of a video transmission system, the transmitter accepts the two original left and right images as input, and computes a disparity map. This map is then transmitted to the receiver along with the reference perspective view. Using this information, the receiver may then "synthesize" the appropriate intermediate view according to the demands of the viewer.

Before going any further, let us examine how the intermediate view reconstruction problem has been approached in the past.

## 3.2 Methods of intermediate view reconstruction

### 3.2.1 3-D model-based vs. 2-D signal processing techniques

I wish to underline at this point that a clear distinction needs to be made between IVR using 3-D models (typically used in the domain of computer vision), and IVR using 2-D signal processing. In computer vision applications, cameras typically have a very large interaxial separation, and utilize the toed-in configuration with a large convergence angle. The cameras take snapshots of *objects* from very different angles in order to completely describe every point of the object (except perhaps the underside which is not seen). The goal is to develop a 3-D model of the object which will then be used to perform coordinate transformations and to display the object from arbitrary viewing angles. This is more generally called the Arbitrary View Generation (AVG) problem, and applications are in the area of robot vision, virtual environments, and 3-D object modeling.

The approach to solving the correspondence problem, in this case, is quite different. The various snapshots are *registered* to form one complete description of the object. The registration process is a complicated task since there may not be high

correlation between the images, and since occlusions pose a serious problem. To further complicate the task, additional luminance imbalances often exist in the images since cameras are located in completely different coordinate systems (due to different illumination, shadowing, etc ... ). The process of view registration is an area of active research today. Here, it becomes important to be able to manipulate objects in a 3-D scene so as to depict realistic views from any arbitrary viewing angle.

The significant advantage with the 3-D model-based approach is that once a model for an object is computed, there is great flexibility in manipulating and transforming the object. The generation of arbitrary views is easier to accomplish, and not only to accommodate *lateral* viewer head movements, but for any head movement whatsoever. Furthermore, arbitrary views can be computed and displayed in real-time with today's high performance workstations[2]. However, sophisticated graphics software and 3-D graphics hardware accelerators are required for visualization. In addition, the execution times of such algorithms are dependent on the complexity of the scene (object).

In contrast, there is the arbitrary view generation problem in the context of 3-D for entertainment purposes (e.g., IMAX 3-D films). Here, realistic perspectives of a scene are required only within a limited range (in the vicinity of the original cameras), and hence the task is typically labeled "intermediate view reconstruction" rather than "arbitrary view generation". 2-D signal processing techniques are used to generate intermediate pictures since computing a model for any arbitrary real-world scene is not possible today[3]. No assumption is made about the physical scene that has been captured, and no attempt at computing $2\frac{1}{2}$-D surfaces is made. For image reconstruction, interpolative techniques based on the 2-D image data given by the original perspectives are used.

In 3-D for entertainment purposes, stereoscopic cameras are typically arranged in the parallel (or near-parallel) configuration, and have a relatively small interaxial

[2]The "Virtual Environment" room at the National Research Council of Canada (NRCC), for example, has two high-speed personal computers which compute arbitrary views of an object in real-time according to the viewer position returned by a head-tracker.

[3]In fact, even with computer vision techniques that model objects, often certain assumptions as to the size and shape of the object are made in order to simplify the problem. The major disadvantage with the 3-D model-based approaches is that they are not capable, today, of modeling (hence reconstructing) complex real world scenes.

separation. Hence, there is a high correlation between the left and right images. In a typical application, only a few perspective views need to be computed. However, these techniques do address the problem of IVR for complex real-world scenes, and hence are very useful for applications where computer vision techniques are unsuitable, such as parallax adjustment. Furthermore, such techniques have the property that execution times are independent of scene complexity, and do not require graphics accelerators. On the other hand, they require high processing power and large amounts of memory.

### 3.2.2 3-D model-based techniques

There are many publications on AVG for 3-D objects in the literature [4, 10]. This section offers a brief article summary on the topic.

In [11], Chang and Zakhor discuss the task of arbitrary view generation from images acquired from an uncalibrated video camera[4]. To acquire images, a camcorder is translated across the object following a straight line; the generation is repeated at different elevations ("View 1" and "View 2") as shown in Figure 3.2.

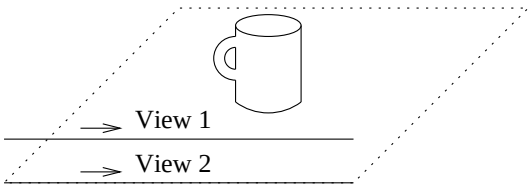



**Fig. 3.2** Experimental setup used for image acquisition in the context of arbitrary view generation in [11].

Certain reference locations (particular views of the object) are chosen for which depth information is sought. To do this, disparity is estimated between these reference frames and their closest neighbours. For every point $(i, j)$ in the reference image, depth is estimated as the inverse of disparity $d(i, j)$ found from exhaustive search block-matching (Section 3.4.1) along the epipolar line. Searching for scalar disparity vectors is consistent with the parallel-optics camera assumption, and estimating depth as the inverse of disparity is consistent with relationship (2.7) for parallel cameras. In

[4] An uncalibrated video camera is one whose position relative to real world coordinates is unknown. Camera calibration is helpful at determining the real world coordinates of a point, given its relative coordinates in the acquired images.

fact, the parallel camera assumption here is reasonable since the camcorder motion is translational only, not rotational.

After the initial disparity estimation, confidence levels are assigned to different regions in the reference image. Properties considered in this process are aperture ambiguity (matching of a block which contains insufficient variation or texture), constant intensity regions, occlusions, and inconsistencies in matching resulting from comparisons of depth maps from the various neighbours. To deal with some of these problems, an adaptive approach is utilized whereby constant intensity regions are handled with larger block sizes, and boundary regions with smaller block sizes. Depth maps between the reference frame and its closest neighbours are then normalized and combined to form a single depth map (points are combined in a weighted average based on confidence levels). The remaining low confidence regions are interpolated from neighbouring high confidence regions.

The end result of this process is a set of estimated depth maps, one for each chosen location around the object. The final step is to estimate the relative camera motion between all reference frames so as to be able to select the appropriate one(s) needed for reconstruction. This results in a geometric relationship which ties together the different reference frames.

To reconstruct an arbitrary view given a viewpoint in real world coordinates, the appropriate reference frame(s) are chosen. Initial estimates are obtained by applying the computed geometric motion parameters to each of the chosen reference frames. The estimates are then combined into a single image, interpolating where necessary. The details of implementation of each of these stages are left for the reader to explore in [11].

Once the costly task of computing a model for the 3-D object is executed, it becomes relatively simple to compute any arbitrary view. The results presented in [11] are good-quality reconstructions of a stationary 3-D object at different "virtual" positions. However, this technique is not well-suited for complex scenes where many depth planes exist and where only two (or a few) slightly displaced perspective views of the scene are available. For this, 2-D signal processing techniques offer an interesting alternative.

### 3.2.3 2-D model-based techniques

All publications on intermediate view reconstruction using 2-D signal processing consider the correspondence problem, i.e., disparity estimation, to be the most important task. The accuracy of the disparity estimation has a direct impact on the quality of the reconstructed view. For example, in both papers from Skerjanc and Liu relating to IVR, [12, 13], the vast majority of the content is used to describe the disparity estimation algorithm. There is only a small paragraph describing how the view reconstruction is done. Nevertheless, in this section I would like to briefly look at the some existing technologies for reconstructing intermediate images.

In [12], Skerjanc and Liu use the familiar parallel camera model, where a point in the real world coordinate system projects onto two different image planes at the points $(x_l, y_l)$ and $(x_r, y_r)$. This model was examined in some detail in Section 2.2.2, where a simple equation describing disparity was derived. The relation between the horizontal coordinates in the left and right images, $x_l$ and $x_r$ is given by

$$x_l - x_r = 2h - \frac{fb}{z}, \tag{3.1}$$

where

$f$ = camera focal length,
$b$ = baseline distance,
$h$ = CCD sensor shift, assuming horizontally-shifted cameras with parallel optics,
$z$ = depth of point $P$ in the real world coordinate system.

Equation (3.1) forms the basis for the intermediate view reconstruction. For a virtual camera located at a baseline distance of $b' < b$ from the left camera, equation (3.1) gives an equation for calculating the $x$-coordinate of the homologous point, $x_i$,

$$x_i = x_r + 2h - \frac{fb'}{z}. \tag{3.2}$$

Assuming a negligible CCD camera shift distance (i.e., $h \to 0$), but still maintain-

ing the parallel camera assumption, (3.1) and (3.2) give:

$$x_i = x_r + \alpha(x_l - x_r), \tag{3.3}$$

where $\alpha$ is the normalized distance of the virtual camera position with respect to the left camera (i.e., $\alpha = \frac{b'}{b} \in [0, 1]$). With the parallel camera configuration, there is no vertical parallax, and $y_l = y_i = y_r$. Thus, having solved the correspondence problem, one can easily synthesize an intermediate view at the position $\alpha$ using equation (3.3), by shifting the picture point $(x_r, y_r)$ by the scaled known disparity value, $\alpha(x_l - x_r)$.

In [14], Siegel *et al.* also stress the importance of a good disparity estimator for intermediate view reconstruction. They highlight that the task of disparity estimation is inherently noisy due to *occlusions* and *correspondence ambiguity* caused by periodic structures in the image. These phenomena complicate the correspondence problem since either *no corresponding points*, and *too many corresponding points*, respectively, can be found. Occlusions occur when foreground objects block the visibility of background objects at greater depths, and the aperture problem poses problems in textureless or periodic structures, where ambiguous matches are found.

To reconstruct high quality intermediate views, Siegel *et al.* concentrate on the computation of a reliable disparity map. The authors try to deal with occlusions and aperture ambiguity by identifying, and then correcting, such problem areas in the image. To do this, three important observations are made:

1. Given a stereoscopic pair of images, a disparity vector describing a token in the left view is equal to the negative disparity vector of the homologous token in the right view. They use this fact to flag and discard contradictory disparity vectors as unreliable.

2. A discontinuity in a disparity map along an epipolar line indicates an occlusion. This observation is used for the detection of occluded regions.

3. The reliability of a disparity estimate for a contiguous region of pixels can be approximated by the inverse of the PSNR of the image region, where the noise is defined as the error between the original region and the corresponding region in the reference image.

An initial disparity map is obtained via block matching (Section 3.4.1). Observations 1 and 3 above are then applied, and erroneous estimates are eliminated. Next,

based on observation 2, occluded pixels are tagged, and their disparity values adjusted based on neighbouring, unoccluded pixels.

The result is a more accurate disparity map than that obtained initially from simple block matching. To generate an intermediate view, the intersection of each pixel's disparity vector with the intermediate image is computed. The corresponding pixel's intensity value, from either the left or the right view, is then mapped to the intermediate image at that point. It is interesting to note that with this approach, the pixel's intensity value is taken from only one perspective view, which assumes a perfect match given by the disparity vector. Alternatively, one could take a weighted average of the left and right image points. The issue of how to reconstruct a token once a match is found (i.e., from one or both images), is discussed in more detail in Section 5.1.

This method for intermediate view reconstruction gives no guarantee that the intersections of disparity vectors with the intermediate image plane will fall on grid points of the original sampling lattice. Therefore, advanced techniques are required to interpolate sample points from a non-uniform grid. This is a complex procedure, and the authors do not indicate the approach taken. The authors do, however, show very interesting image reconstructions obtained for the monoscopic test sequence, "flowergarden". The disparity to predict frame 3 from frame 0 was computed and used to interpolate frame 1 of the test sequence. Comparing the reconstruction with the original frame 1, a PSNR of 28.14dB was reported.

In [15], a technical report from the Cambridge Research Lab, several other so-called image-based (2-D model-based) rendering techniques are presented. These techniques are such that they rely primarily on the original set of images to produce virtual views. One such class of techniques is the *non-physically based image mapping* technique often used in the advertising and entertainment industries. This technique, an example of which is feature-based morphing, performs image correspondence between a pair of possibly unrelated images. A set of oriented lines is manually selected by the user in both the source and target images. The morphing process then performs warping of the two images so that the source shape slowly takes the form of the target shape. To do this, pixels from each manually designated line in the target image are mapped to their corresponding lines in the source image. Although this technique offers great

flexibility to the animator with respect to the choice of feature correspondences, it is computationally intense.

Another class of image-rendering techniques discussed in this technical report is the *mosaicking* approach. Here, two or more images taken from different viewpoints of a scene are combined to form one larger image with a wide field of view. Intermediate views are contained within this larger image, and from it, an arbitrary view can be quickly generated. In order to combine constituent images, image registration must be performed, and there are a multitude of techniques for doing this. The main concern in created the mosaic is to minimize distortions at the seams where constituent images connect.

Finally, there is the class of techniques called *interpolation from dense samples*. The idea here is that many image samples of an object or scene are acquired, and, based on these, some form of a lookup table is generated. The stored lookup table is then used to interpolate data for an arbitrary view. The significant advantage of this method is that it does away with the need to solve the correspondence problem, which is a very complicated, time-consuming task. This technique offers rapid image synthesis of intermediate views since data is acquired from a lookup table, which translates to fast visualization speeds. However, it suffers from high storage and memory requirements, as well as the knowledge of the camera viewpoints at every sample.

## 3.3 Correspondence problem and disparity estimation

Stereoscopic *disparity* is defined as the physical distance in position between homologous points of a stereo pair, i.e., points resulting from the projection of some 3-D point onto two image planes. Hence, disparity is a vector describing how a token (region, block or pixel) translates from one image to the other. Given a stereo pair of images, one the *reference*, the other the *current* image, *disparity estimation* is the process of grouping every token (pixel, block, region, etc ... ) in the current image with its corresponding token in the reference image. The output of a disparity estimator is therefore a vector field which shows the two-dimensional displacement of each token in the current image is displaced with respect to the reference image.

The stereo matching problem can be represented mathematically as a minimization. Given two images, $I_{ref}$ and $I_{cur}$, for each token $i$ in the reference image, we find the disparity vector, $\hat{d}_i$, which minimizes some cost function, $U(\cdot)$[5], i.e.,

$$\hat{d}_i = \arg \min_{d_i} U(I_{cur}(i), I_{ref}(i + d_i)). \tag{3.4}$$

Assume that $\bigcup i, \forall i = I_{cur}$ (i.e., the union of tokens gives the current image). Then, once $\hat{d}_i$ is determined for each $i \in I_{cur}$, we are left with a vector field which completely describes the current image *in terms of the reference image*.

The applications where disparity estimation is essential are numerous. First and foremost, it allows for redundancy elimination and hence efficient stereoscopic video compression. Rather than transmitting two independently-compressed images, one can transmit the reference image plus the vector field only, which results in substantially less information to send since cross-image correlation is exploited. The receiver can then reconstruct the current image from the decompressed reference image and vector field. The MPEG standard for video compression accommodates what is called an "auxiliary stream" which carries a disparity map as well as disparity-compensated prediction error for the view to be reconstructed. The "main stream" is used to transmit the reference image.

For efficient stereoscopic video compression, the properties of the vector field itself must be exploited since they directly affect the rate allocated to disparity information. In this work, since we are mainly using the disparity field as a means for image reconstruction, we are not considering the properties of the vector field since compression is not our main goal.

Intermediate view reconstruction is another application which disparity estimation makes possible. In order to reconstruct an image at some arbitrary position, first the correspondence problem must be solved. The idea is to use the mapping between the left and right images to interpolate some intermediate view. The task of *disparity estimation* is the essential problem which we must first solve before performing intermediate view reconstruction. Therefore, the next section is devoted to existing disparity estimation algorithms, with the intention of adapting them to the

[5]The selection of an appropriate cost function will be discussed in Section 4.3.6.

intermediate view reconstruction problem. The methods examined in the following section deal primarily with estimation for disparity-compensated prediction of one perspective view in terms of the other.

## 3.4 Methods of disparity estimation

The problem of disparity estimation resembles, to a large extent, that of motion estimation, of which there are many well-known algorithms and publications. The problem of motion estimation is mainly used for redundancy elimination in sequences of images, where motion-compensated prediction leads to efficient video compression. Motion estimation methods can in fact be applied to disparity estimation, since the only fundamental difference is that motion estimation considers images taken at different times, and disparity estimation deals with images taken at the same time, but from different perspectives. However, a disparity field has distinguishing features which must be taken into account.

Firstly, a disparity vector, under the assumption of parallel cameras, or after suitable image rectification, is a scalar (i.e., no vertical component), whereas motion is typically a 2-D vector. Secondly, the dynamic range of a disparity vector is typically larger than that of a motion vector; horizontal screen parallax values of 25-30 pixels are not uncommon for a stereoscopic pair. Thirdly, for motion sequences, there is temporal continuity of motion, which does not extend to disparities in practice. That is, motion sequences, which have dense temporal sampling (e.g., every 1/60-1/50s), will have similar motion vector fields between frames. For disparities, one can think of a multi-view system where several cameras are displaced by a distance of $\delta\alpha$ laterally from one another, capturing the same scene. As $\delta\alpha \rightarrow 0$, then neighbouring images will have similar disparities. In practice however, cameras may not be placed infinitely-close to each other due to physical limitations, and so the disparity continuity analog to temporal continuity is quite weak. Finally, while motion-induced parallax is due to the combination of both object and camera motion, disparity-induced parallax is conceptually simpler since it is due purely to a simple shift (and perhaps a small rotation) of the camera.

### 3.4.1 Block-based approaches

Block-based disparity estimation algorithms select as the matching token a block of arbitrary size. In this case, the correspondence problem is reduced to grouping together blocks of pixels between the reference and current images; this is called the block-matching (BM) algorithm. The resulting vector field assigns the same disparity vector to all pixels of a block, resulting in a sparse vector field.

In the context of BM, a popular choice for the cost function of in (3.4) is the absolute value function. This approach is called the minimum mean absolute difference (MAD) approach, and its popularity stems from the fact that it lends itself nicely to VLSI implementations. This is one reason why it is preferred over the minimum mean square error (MSE) function since the square operation is more complex to realize in hardware. A more thorough discussion on the selection of an appropriate cost function is left for Section 4.3.6.

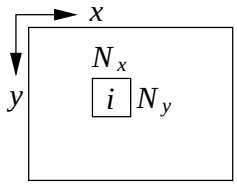



**Fig. 3.3** Coordinate system used for block matching. The current (predicted) image is broken up into blocks of size $N_x \times N_y$.

First, we establish a coordinate system as shown in Figure 3.3. The current image is broken up into blocks of size $N_x \times N_y$. We start with the parallel-camera assumption and only allow a horizontal disparity component (i.e., epipolar lines are parallel to the scan lines of the images). Then, for each block $i$ of the current image, and given a set of candidate disparity vectors, the optimal vector $\hat{d}_i$ is the one which minimizes the following equation:

$$\hat{d}_i = \arg \min_{d_{i_x}} \frac{1}{N_x N_y} \sum_{(m,n)\in\beta_i} |I_{ref}(m + d_{i_x}, n) - I_{cur}(m, n)| , \tag{3.5}$$

where $\beta_i$ denotes the $i$'th $N_x \times N_y$ block of the current image, and $I_{ref}$ and $I_{cur}$ denote the reference and current images, respectively. A typical way of establishing the set

of candidate disparity vectors is to simply define a state space around block $i$, and select every vector in that state space as a candidate. The selection can be done at full- or sub-pixel precision. This is called the *exhaustive search* approach, and is very well known. In (3.5), the term $\frac{1}{N_x N_y}$ can be left out in computation since is does affect the minimization.

The major problem with the block-based approaches is that they result in sparse vector fields which do not accurately describe all pixels in the image. However, block-based approaches are desirable due to their simplicity, and due to the fact that they lend themselves naturally to parallel processing and hardware implementation. There is another block-based disparity estimation technique, however, which does result in dense vector fields. The method is based on "sliding blocks" and has been described in in [16].

Ben Slima, Konrad and Barwicz improve the sliding-block approach by making a simple observation in [17]. In the standard approach with sub-pixel precision, implicit low-pass filtering is applied to one image of the stereo pair via interpolation. This results in an imbalance between the images and could bias the disparity estimation. Instead, the authors apply the same filtering to both images, i.e., *balanced filtering*, and hence the same noise suppression is applied to both perspective views. A better disparity estimate over the whole image is therefore expected. For right-image prediction of test sequences "train" and "manege", the authors show a rise of up to about 2dB in peak prediction gain $(PPG)$[6] as compared to the traditional sliding blocks technique.

### 3.4.2 Pixel-based approaches

The computation of motion or disparity is an ill-posed problem since the existence, uniqueness, and stability of solutions cannot be guaranteed in the absence of additional constraints. Although this is true for block-based approaches as well, it is a more serious threat in the pixel-based approaches. Typically, regularization theory is used whereby an additional smoothness constraint is used to restrict the space of

[6]The $PPG$ is defined as $10 \log_{10} \frac{255^2}{\mathcal{E}}$, where $\mathcal{E}$ is the mean-squared prediction error of the right image $I_R$, $\frac{1}{K} \sum_{(x,y)} [I_R(x,y) - \hat{I}_r(x,y)]^2$. The PPG measure is used in Chapter 5 to gauge the performance of algorithm modifications.

acceptable solutions to *smooth* vector fields.

March has adapted this technique to the disparity estimation problem [18]. He imposed a smoothness constraint on the disparity vector field under the basic physical assumption that the coherence of matter tends to give rise to smoothly varying characteristics in a real-world scene. The ill-posed disparity estimation problem then becomes well-posed, and the additional smoothness constraint measures the degree of regularization (or smoothness) of the solution.

The computation of disparity consists of the minimization of a penalty functional, $\mathcal{P}(u)$, which measures the discrepancy between the solution and the input data, plus a regularization or smoothness term, $\mathcal{R}(u)$, where $u(x, y)$ is the disparity function. A multiplier, $\lambda$, is introduced to control the compromise between the closeness of the solution to the data and the degree of smoothness.

The minimization in question is carried out over all tokens (pixels, in this case) in the current image (left - $I_L$), and results in a dense disparity map, i.e., one vector per pixel. It is given by the following formulas.

$$\min_u \mathcal{E}(u) = \sum_{(i,j)\in I_L} \mathcal{P}(u) + \lambda \mathcal{R}(u), \tag{3.6}$$

where

$$\begin{aligned} \mathcal{P}(u) &= \left[I_L(i,j) - I_R(i + u(i,j), j)\right]^2, \\ \mathcal{R}(u) &= \left[(u(i,j) - u(i,j-1))^2 + (u(i,j) - u(i-1,j))^2\right]. \end{aligned} \tag{3.7}$$

To find a solution, we need:

$$\frac{\partial \mathcal{E}(u)}{\partial u(i,j)} = 0, \quad \forall\ (i,j). \tag{3.8}$$

The left and right images, denoted by $I_L$ and $I_R$, are luminance components only, and the parallel camera assumption is made such that the disparity vector is a scalar with a horizontal component only. In this example, the right image is the reference image, and the left is the current image.

Expanding (3.8), we get

$$-\left[I_L(i,j) - I_R(i+u(i,j),j)\right] I_{R_x}(i+u(i,j),j) + \lambda(4u(i,j) - u^*(i,j)) = 0, \tag{3.9}$$

where $I_{R_x}$ denotes the partial derivative of $I_R$ with respect to $x$, and

$$u^*(i,j) = \frac{1}{4}(u(i-1,j) + u(i,j-1) + u(i+1,j) + u(i,j+1)), \tag{3.10}$$

is the local average of the disparity vector at the point $(i,j)$. Solving for the disparity, we get

$$u(i,j) = u^*(i,j) + \frac{1}{\lambda}\left[I_R(i,j) - I_R(i+u,j)\right] I_{R_x}(i+u,j), \tag{3.11}$$

where a factor of 4 has been incorporated into $\lambda$.

March uses Gauss-Seidel relaxation to solve (3.11), but modifies it slightly to improve the numerical stability of the algorithm. As a result, with $n$ as the iteration number, the final solution, that can be implemented in a computer simulation program, is given by:

$$u^{n+1}(i,j) = u^{*n}(i,j) + \frac{1}{\lambda}\left[I_L(i,j) - I_R(i+u^{*n},j)\right] I_{R_x}(i+u^{*n},j). \tag{3.12}$$

For the current iteration $(n+1)$, $u^{n+1}(i,j)$ is based on $u^{*n}(i,j)$, a local average of $u(i,j)$ from the *previous* iteration.

March argues that the local average, $u^{*n}$ as defined in (3.10), destroys disparity discontinuities at object boundaries since vector smoothing is done across these boundaries. A better way of reconstructing boundaries is to smooth only *within* objects. Of course then the problem is to find where the boundaries are in the image. This is a non-trivial problem, and one which is not explored in the paper. March assumes a priori knowledge of the boundaries, and uses a controlled-continuity stabilizer which deactivates smoothing across boundaries.

Note that since there is no guarantee that $u^{*n}$ will be an integer, picture points

which do not fall on the original sampling lattice are required. Section 4.6 discusses the interpolation function used to acquire these points.

### 3.4.3 Other approaches

There are many other publications on methods of disparity estimation. To reference just a few of them, there is joint motion/disparity estimation in [19, 20, 21], segmentation-based coding in [22, 23], and Maximum A Posteriori (MAP) estimation in [24]. Next-generation region-based stereoscopic video coding is found in [25]. Most of these schemes are advanced algorithms which, in the context of this project, have not been explored vis-à-vis intermediate view reconstruction.

# Chapter 4

# Disparity estimation for intermediate view reconstruction

Existing techniques for disparity estimation have been looked at in the context of disparity-compensated predictive coding, where one perspective view is coded with respect to the other. The problem of intermediate view reconstruction (IVR) has also been looked at. This chapter discusses disparity estimation in the context of the reconstruction of intermediate views, and presents novel 2-D model-based approaches to solving this problem.

Two of the existing disparity estimation techniques, examined in Chapter 3, are adapted to the IVR problem. For each of these methods, the focus is on obtaining the most accurate disparity field possible between perspective views in order to obtain high-quality reconstructions.

The chapter begins with a presentation of the underlying model used for reconstructing intermediate views. The BM algorithm, which was presented in Section 3.4.1, is adapted to IVR. Major improvements to simple block-matching are proposed. The impact of each incremental improvement on the quality of the estimated correspondence, in the form of disparity vector fields, is presented. The pixel-based regularization approach to disparity estimation is also adapted to IVR, and a small but important improvement is proposed; once again, results of disparity estimation are presented. Finally, a technique which improves both the algorithm execution time and the quality of output, and which is common to both the block- and pixel-based

approaches, is presented.

## 4.1 Stereoscopic test sequences

Estimation algorithms are applied to various stereoscopic test sequences (left- and right-eye views) originally captured by CCETT, Rennes, France for the RACE 2045 - DISTIMA European project, and by the NHK, Japan. The RACE - DISTIMA project is involved with the development of a system for capture, coding, transmission and presentation of digital stereoscopic image sequences[1].

Test sequences "*flower*" from NHK, and "*piano*" from CCETT were used for testing within this project. Both are interlaced sequences in the YUV colour-space, and have chrominance components subsampled horizontally by a factor of 2 with respect to luminance (format "ITU-R 601", "YUV" colour-space, 4:2:2). In terms of image content, test sequence "*piano*" is simpler, with few depth planes and occlusions. Four major objects define the depth planes: the piano player, the piano, the flowers and the background. Test sequence *flower* is more complicated, with many more occlusions. Figure 4.1 shows field 0 of the original test sequences used, for both left and right perspective views. Note that images are presented with correct aspect ratio using vertical interpolation to fill in the odd field.

Sequences have been converted and stored in "ViDS" format developed at INRS[2]. Test sequence *piano* is 720x576 in dimension, and *flower* is 720x480. Processing is done on the first *field* of each test sequence, which means that only half of the number of lines of one full frame are processed.

## 4.2 Underlying model for IVR

The approaches at estimating disparity presented in this chapter are fundamentally different. However, the underlying 2-D model which describes how the intermediate view reconstruction is carried out, is the same. In both cases, the estimation process

[1]See <http://www.tnt.uni-hannover.de/project/eu/> for more information on the RACE European projects.

[2]See INRS VisCom internal document [26] for more information on the ViDS format used within the INRS Visual Communications group.

(a) *flower* - left

(b) *flower* - right

(c) *piano* - left

(d) *piano* - right

**Fig. 4.1** Original stereoscopic test sequences used, field 0. Images are vertically interpolated to fill the odd fields and maintain the correct aspect ratio.

constrains candidate disparity vectors to those that intersect the desired coordinate position in the intermediate image. Figure 4.2 depicts this idea, where each token at position $(i, j)$, in the intermediate image $''I''$, is intersected by a disparity vector $\underline{d} = \underline{d}_{ij}$. Position $(i, j)$ in the intermediate image is a pivot point for candidate token-pairs in the left and right images. A token is defined to be a region or block of pixels (or even a single pixel), such that their union defines an entire image. Only scalar vectors are considered for now; 0 vertical disparity is assumed.

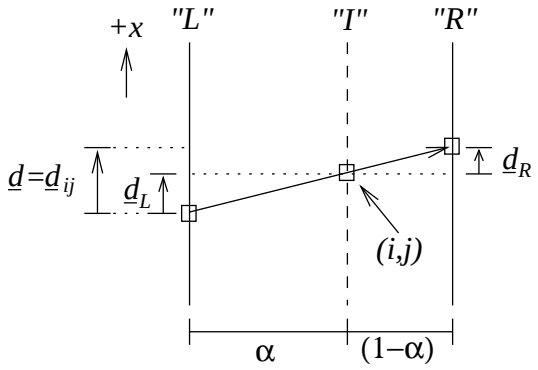



**Fig. 4.2** Top view, from left to right, of the left ($''L''$), intermediate ($''I''$), and right ($''R''$) image planes. Disparity vector $\underline{d}_{ij}$ originates from $''L''$, intersects $''I''$ at position $(i, j)$, and terminates at $''R''$.

The disparity vector is defined in the intermediate image plane, i.e., it intersects a pixel position in $''I''$, but not necessarily in $''L''$ and $''R''$. As indicated in the figure, which shows a top view of the three image planes, vectors going from left to right, are defined as positive in the plane of $''I''$. A frontal view of the model in the plane of the reconstructed image $''I''$ is shown in Figure 4.3. Here, the intermediate image is broken up into blocks, and $\underline{d}_{ij}$ points to estimated homologous blocks in the left and right images. Each pixel in the block at position $(i, j)$ is described by $\underline{d}_{ij}$, and $\underline{d}_L$ and $\underline{d}_R$ are shown at the bottom of the figure.

The position of the intermediate image plane is defined as being at a normalized distance $\alpha$ from the left, reference image. The distance from the left to the right

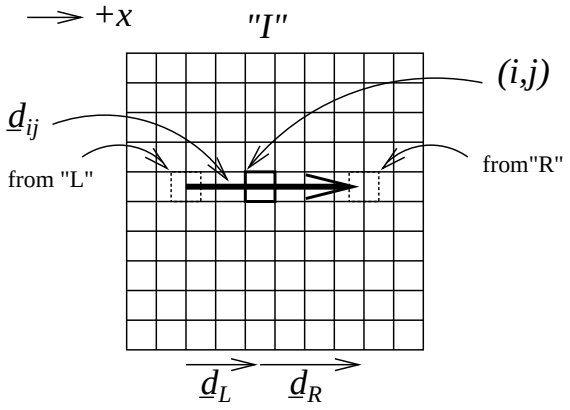



**Fig. 4.3** Underlying IVR model shown in the plane of the intermediate image $''T''$. Token $(i, j)$ in $''I''$ is described by disparity vector $\underline{d}_{ij}$, which points to the corresponding matching token-pair from $''L''$ and $''R''$.

image plane is 1, such that $\alpha \in [0, 1]$[3]. Given a disparity vector $\underline{d}$, from Figure 4.2,

$$\begin{aligned} \underline{d} &= \underline{d}_L + \underline{d}_R, \\ \underline{d}_L &= \alpha \underline{d}, \\ \underline{d}_R &= (1 - \alpha)\underline{d}. \end{aligned} \tag{4.1}$$

After having solved the correspondence problem between images $''L''$ and $''R''$, one obtains a vector for each token in the virtual image pointing to the corresponding homologous tokens positioned at both ends of the disparity vector. This constitutes the basis of our reconstruction model, and if the considered token is image intensity $I$, as is usually the case, then the model can be mathematically expressed as follows:

$$I_I(i, j) = I_L(i + \alpha \underline{d}_{ij}, j) = I_R(i - (1 - \alpha)\underline{d}_{ij}, j) \tag{4.2}$$

The token at position $(i, j)$ of the intermediate view can therefore be reconstructed using the data of the corresponding token from either view, as is done in [13] (see Chapter 3), according to (4.2). Alternatively, it can be reconstructed using a weighted average of the data from the tokens of both views, under the assumption that a *perfect*

[3]In Chapter 5, the possibility of extending this to reconstruct views beyond this range is considered.

match was not found. The advantages and disadvantages of either approach are left for discussion in Chapter 5.

When performing IVR within the framework of this model, it is inherently assumed that all cameras (including virtual cameras) have parallel optics. With $b$ as the baseline distance, the familiar equation describing disparity first derived in Section 2.2.2, $\underline{d} = kb$ ($k$ is a constant, all other things being equal), tells us that any change in the baseline distance induces the same exact change in the length of the disparity vector. It is this that allows us to scale disparity as a function of the intermediate view position, $\alpha$.

The parallel camera assumption is very useful here in that it simplifies the geometry significantly. However, this assumption is only made for view reconstruction, not for disparity estimation as we will see in Section 4.3.2. Here, no assumption is made on the camera configuration.

## 4.3 Block-based disparity estimation

The exhaustive search block matching (BM) algorithm often used in video compression is adapted to perform intermediate view reconstruction. The algorithm is a typical BM, fitted to the model of the previous section. The fundamental difference is how candidate disparity vectors are chosen, based on the intermediate position given by $\alpha$.

### 4.3.1 Adaptation to intermediate view reconstruction

According to the model of Section 4.2, the intermediate image is broken up into an integer number of blocks of a fixed, arbitrary size, $N_x \times N_y$. Then, for each block in the intermediate image, exhaustive search is performed over all candidate disparity vectors. To select the set of candidate vectors for a particular block, a maximum horizontal screen parallax value is chosen, thus defining the search range for the block-matching algorithm[4].

In general, for block $(i, j)$, the set of candidate disparity vectors $\{D_{ij}\}$, is made

[4]For almost parallel camera optics, a maximum total horizontal disparity of 32 pixels is reasonable for the RACE - DISTIMA / NHK test sequences.

up of all vectors within the search range given by $(-d_{\max}, d_{\max})$, at pixel precision $q$ $(\in 1, \frac{1}{2}, \frac{1}{4}, \ldots)$. The token at position $(i, j)$ is a pivot point for all vectors in $\{D_{ij}\}$, i.e., all vectors in $\{D_{ij}\}$ intersect block $(i, j)$ in the intermediate image. For example, at full-pixel precision $(q = 1)$, with $d_{\max} = 32$, there are 33 candidate disparity vectors in $\{D_{ij}\}$. However, with $\alpha$ a real number and $d_L$ and $d_R$ as defined in (4.1), there is no guarantee that vectors will fall on points of the original sampling lattice in either the left or right image planes. The algorithm therefore requires that image intensities which are not directly available as input data be computable from the original picture points.

For each block in the intermediate image, this algorithm searches for the best matching pair of blocks among all candidate pairs defined by the vectors in $\{D_{ij}\}$. To do this, the usual method is employed, i.e., the cost associated with the "difference" between the block-pair defined by each candidate vector is computed, and the one which results in the lowest energy is chosen. A cost function is selected to calculate the difference between a block-pair. Cost function selection is discussed in detail in Section 4.3.6. For now, we represent the minimization in question mathematically for the case of absolute value used as the cost function. Then, for each block $(i, j)$ in the intermediate image (at position $\alpha$), we find the optimal disparity vector $\underline{\hat{d}}_{ij}$ by performing the following minimization:

$$\hat{d}_{ij_x} = \arg \min_{d_{ij_x}} \frac{1}{N_x N_y} \sum_{(m,n)\in\beta_{ij}} \left| I_L^Y(m + \alpha d_{ij_x}, n) - I_R^Y(m - (1-\alpha) d_{ij_x}, n) \right|, \tag{4.3}$$

where $N_x \times N_y$ is the size of the block, $d_{ij_x}$ is the $x$-component of the disparity vector, i.e., $\underline{\hat{d}}_{ij} = [\hat{d}_{ij_x}\ 0]^T$, and $\beta_{ij}$ is the set of pixels in the $ij^{th}$ block. $I_L^Y$ and $I_R^Y$ represent the luminance components of the left and right images, respectively. The term $\frac{1}{N_x N_y}$ in (4.3) can be left out for implementation purposes.

The proposed algorithm has been tested on various sequences using the parameters listed in Table 4.1. Resulting disparity maps for test sequences *flower* and *piano* are shown in Figure 4.4. Disparity maps are based on single fields rather than frames; either the even or odd lines are used only. To maintain the correct aspect ratio, however, disparity maps are vertically interpolated. The maps show full disparity vectors originating from the left viewpoint and terminating at the right viewpoint.

For $\alpha = 0.5$, the midpoint of each vector intersects the intermediate sampling lattice at a grid point. Only sparse disparity fields are shown, i.e., one vector per block.

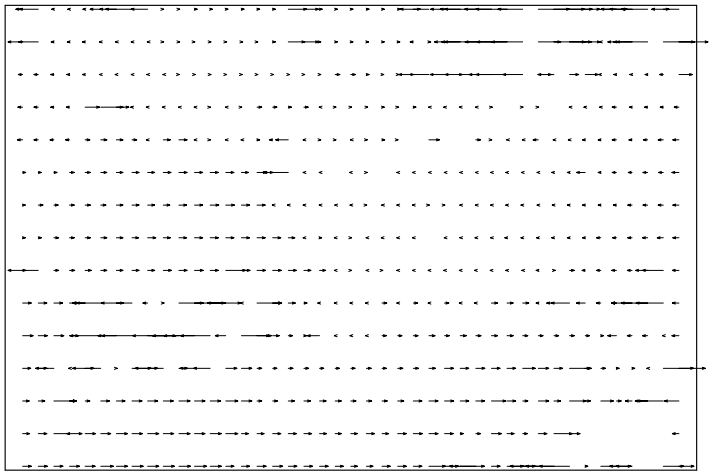
| Parameter | Description | Value |
|---|---|---|
| $\alpha$ | intermediate view position | 0.5 |
| $q$ | pixel precision | 0.25 |
| $\underline{d}_{\max}$ | maximum parallax | (32,0) |
| $N_x$,$N_y$ | block size | 16,16 |

Table 4.1 Parameters of exhaustive-search block matching, used in simulations.

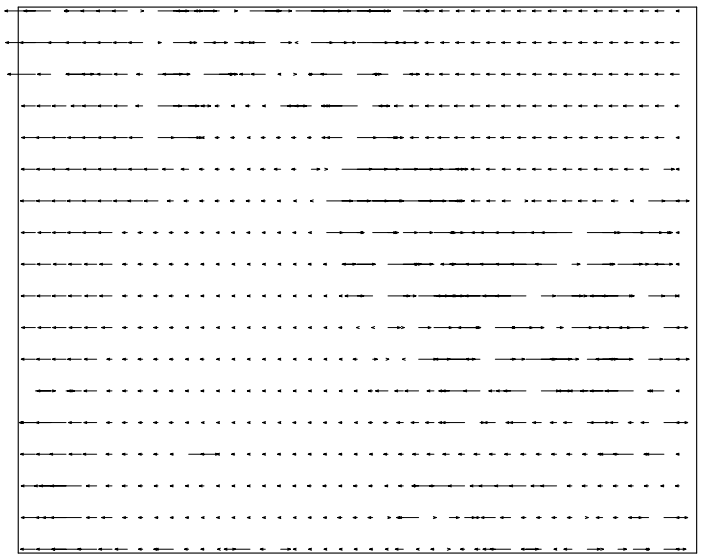
The disparity fields in Figure 4.4 are of a good overall accuracy. For example, in 4.4(b), one can make out the silhouette of the piano player of sequence *piano* (refer to the original image on page 47). The disparity vectors for the object made up of the piano player's body are small compared to those of the background objects. Although this is correct, it seems counter intuitive; we would expect objects at greater depths to have smaller parallax values. However, keep in mind that the images were acquired from slightly toed-in cameras. In this case, the inverse proportionality between depth and disparity, which has been shown to exist only for parallel cameras, only applies to objects which fall *before* the convergence point of the optical axes. For other objects, the inverse is true; i.e., objects at greater depths will have larger parallax values, which is the case here.

The small flower-pot which sits on top of the piano in the top-right corner is also well estimated. Its disparity vectors are slightly larger than those of the piano player's body, since the pot sits in a more distant depth plane. The vector field shows horizontal vector magnitudes of about 9.0 for the flowers, and 3.5 for the piano player's back. These are consistent with "actual" values manually measured from the original data images. Similarly, the disparity field for *flower* in (b) is overall well estimated. The flowers in the large foreground flowerpot on the left have disparity values of 7.0, which is consistent with manually-measured values.

The estimated field for *flower* in 4.4(a) is also quite accurate overall. Here, the zero-depth plane is the brick wall in the background. Hence all objects are located in front of the convergence point of the optical axes, meaning the closer the object, the



(a) *flower*



(b) *piano*

**Fig. 4.4** Estimated disparity fields using exhaustive-search block matching applied to the luminance components of test sequences *flower* and *piano*, field 0. Parameter values used are listed in Table 4.1.

larger its screen parallax (consistent with the inverse proportionality of distance vs. disparity for parallel cameras). This is what one finds when comparing the disparity vectors belonging to the large foreground flowerpot to those belonging to the center of the image, which are much smaller. The flowerpot is mostly described by vectors of length 8, and vectors in the center of the image have horizontal components which are closer to 0.

Nevertheless, the disparity fields in Figure 4.4 have serious inaccuracies. Although the structure of the player's body in 4.4(b), for example, is clear, his head is slightly drowned in the background. This is due to the fact that there is no great luminance detail in the head vs. the background, and ambiguous matches are found. As well, the section of the piano above the keyboard is a low-texture region, and there are great variations in the vectors of this region, while one would expect them to be quite similar. Finally, within the piano player's back, there is one large disparity vector near the bottom which is clearly incorrect; it completely contradicts surrounding vectors which are themselves known to be accurate. This anomaly is perhaps due to the fact that the region has large-scale detail, and the block size used does not capture enough detail so as to find a good match. Rather than using a larger block size which could introduce more serious problems, one way to solve this problem could be to force a likeness of vectors within an object. Similarly in (a), notice the large vectors in the top right corner of the disparity field where one would expect small vectors (indeed 0) since this these vectors are within the zero-depth plane.

The effect that the previous disparity field inaccuracies will have on the reconstructed image will become more clear in Chapter 5. For now, the focus is on obtaining the most accurate disparity fields possible. The approach taken at improving the disparity estimates from Figure 4.4 is to identify problem areas, understand the root of the problems in these areas, and attempt to solve them. The following sections propose incremental improvements to the standard exhaustive-search BM scheme. The end result remains an inherently BM approach at disparity estimation, but offers significant performance gains.

### 4.3.2 Vertical disparity

Test sequences used were originally acquired from cameras which were not exactly parallel. A slightly toed-in camera configuration was used, meaning that vertical parallax due to keystone distortion does exist in the sequences. As already mentioned, vertical parallax further complicates the correspondence problem. One way to deal with vertical parallax is to adopt an *image rectification* pre-processing stage which eliminates, to a large extent, vertical parallax caused by keystone distortion [8].

Alternatively, one could allow for vertical parallax by extending the exhaustive search to two directions, horizontal *and vertical*. Of course, this increases the computational complexity of the algorithm significantly, but it preempts the need to make any special assumptions on the camera configuration as well as the need for any pre-processing stage related to elimination of vertical parallax.

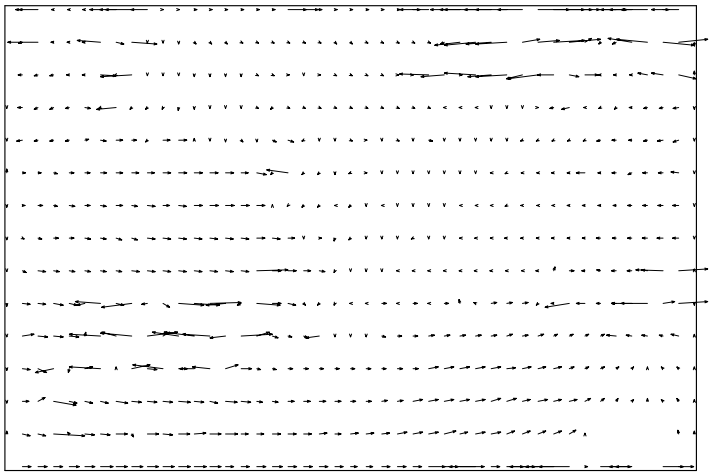
To this end, the minimization is extended to a 2-D search. The chosen disparity vector $\hat{\underline{d}}_{ij}$, for every block $(i, j)$ in the intermediate image (defined by the set of points $\beta_{ij}$), is given as

$$\hat{\underline{d}}_{ij} = \arg \min_{(d_{ij_x}, d_{ij_y})} \sum_{(m,n)\in\beta_{ij}} \Big| I_L^Y(m + \alpha\, d_{ij_x},\, n + \alpha\, d_{ij_y}) - I_R^Y(m - (1-\alpha)\, d_{ij_x},\, n - (1-\alpha)\, d_{ij_y}) \Big|. \tag{4.4}$$

Notice the inclusion of a vertical disparity component in the minimization.

For the exhaustive-search approach, it is important to set a reasonable 2-D maximum disparity value, $\underline{d}_{\max}$; this defines the search range. I have found $\underline{d}_{\max} = (32, 2)$ to be appropriate for the chosen test sequences. Other parameter values used are listed in Table 4.1, and the resulting estimated vector field for test sequence *flower* is shown in Figure 4.5.

The addition of a vertical search in the minimization is successful in correctly matching areas which contain vertical parallax in the images. For example, for test sequence *flower*, there is about a one-line vertical parallax value for the thick pipe on the brick wall in the top right corner of the image (see Figure 4.1(a),(b)). The estimated disparity vectors in this region of the image, shown in Figure 4.5, do in fact have the correct vertical magnitudes. In the case of the estimated field for *piano*,



(a) *flower*

**Fig. 4.5** Estimated disparity field using 2-D exhaustive-search block matching applied to the luminance component of test sequence *flower*, field 0. Parameter values used are listed in Table 4.1, with a maximum allowable vertical parallax of 2 lines.

changes induced by the inclusion of a vertical search are mainly in areas of low texture. This is to be expected since the increased search range will result in lower local minimums in certain regions.

### 4.3.3 Luminance and chrominance imbalances

Stereoscopic film makers are very careful to use cameras with almost identical parameters. As a general rule, the two perspective views should be as well matched as possible. Imbalances in focus, luminance and colour between the views could cause viewer discomfort, and this is a situation which film makers understandably try to avoid.

In the context of disparity estimation, there is a more important reason why perspective view imbalances need to be avoided. The fundamental assumption behind the correspondence problem is that homologous points have the same characteristics, i.e., luminance and chrominance values, since they are projections of the same point in 3-D space. However, this is often not the case for precisely the reason of camera parameter mismatches. If this fundamental assumption underlying the correspondence problem, on which disparity estimation is also based, does not hold, then disparity estimation results may be unreliable.

A balance compensation scheme has been proposed in [27, 28] which attempts to eliminate luminance imbalances between stereoscopic pairs of digital images. Appropriately, the basic assumption of the approach is that imbalances are due to unequal camera parameters only, i.e., lighting conditions are considered to be the same for the two viewpoints.

The idea is to perform a simple linear transformation on the luminance component of the right perspective view $I_R^Y$, in order to equate its mean and variance to that of the left view, $I_L^Y$. It is a pre-processing stage which yields a transformed right view $I_{R'}^Y$. That is (omitting superscript "$Y$" for now),

$$I_{R'}(i,j) = aI_R(i,j) + b, \qquad \forall\ (i,j) \in I_R. \tag{4.5}$$

The mean $\mu_{R'}$, and the variance $\sigma^2_{R'}$, of the transformed right view are thus given by

$$\begin{aligned} \mu_{R'} &= a\mu_R + b, \\ \sigma^2_{R'} &= a^2\sigma^2_R. \end{aligned} \tag{4.6}$$

Solving for the parameters $a$ and $b$,

$$\begin{aligned} a &= \frac{\sigma_{R'}}{\sigma_R}, \\ b &= \mu_{R'} - a\mu_R. \end{aligned} \tag{4.7}$$

As mentioned, the two constraints we need to satisfy are $\sigma_{R'} = \sigma_L$ and $\mu_{R'} = \mu_L$. Therefore,

$$\begin{aligned} a &= \frac{\sigma_L}{\sigma_R}, \\ b &= \mu_L - \frac{\sigma_L}{\sigma_R}\mu_R. \end{aligned} \tag{4.8}$$

Substituting (4.8) into (4.5), the final form of the linear transformation applied to the right perspective view $I_R$, and yielding the transformed view $I_{R'}$, is given by

$$I_{R'}(i,j) = \frac{\sigma_L}{\sigma_R}\Big(I_R(i,j) - \mu_R\Big) + \mu_L, \qquad \forall (i,j) \in I_R. \tag{4.9}$$

The authors balance only the luminance components of the two viewpoints using this technique, since it is the component with the greatest influence on the disparity estimation [27]. However, our test sequences have noticeable colour imbalances as well, particularly test sequence *flower*. The proposed balancing technique has therefore been applied to the colour components as well, yielding a set of $a/b$ parameters for each component. Table 4.2 lists the experimentally computed parameters for various RACE - DISTIMA / NHK test sequences. Listed parameters are averages of the parameters found from the first twenty fields of each test sequence, where scene-cuts were known not to exist. Original sequences *manege* and *piano* had the largest component differences between views.

Although local variations persist, the balance compensation algorithm is very suc-

| | Y | | U | | V | |
|---|---|---|---|---|---|---|
| **Sequence** | $a$ | $b$ | $a$ | $b$ | $a$ | $b$ |
| *flower* | +1.076 | −8.345 | +1.049 | +0.139 | +0.981 | −0.388 |
| *manege* | +1.174 | −2.771 | +1.169 | −0.769 | +1.327 | −0.574 |
| *piano* | +1.109 | +1.065 | +1.237 | +0.161 | +1.081 | −1.204 |
| *train* | +1.041 | −5.970 | +1.068 | +2.132 | +0.939 | +0.638 |
| *tunnel* | +1.029 | −5.437 | +1.118 | +1.851 | +0.998 | +0.709 |

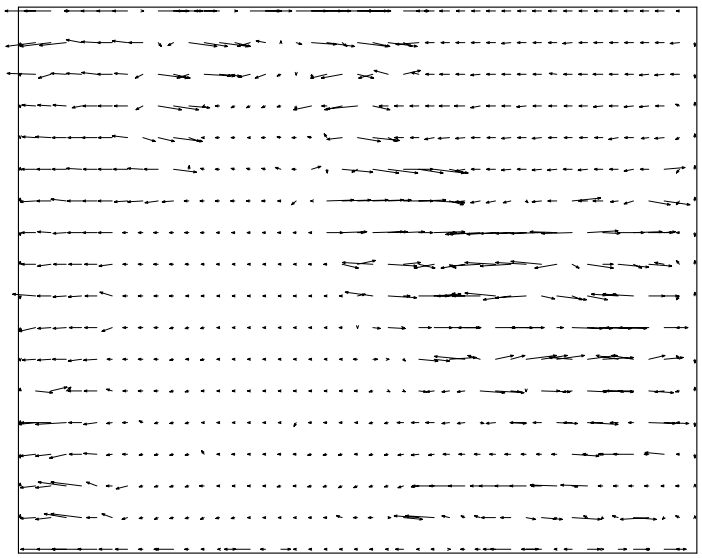
Table 4.2 Luminance and chrominance balance compensation algorithm; computed parameters of linear transformation $ax+b$ on right perspective view of various RACE - DISTIMA / NHK test sequences. Parameter $a/b$ pairs shown for each image component.

cessful at balancing the *global* luma and chroma signals of the perspective views. The balanced image pairs are not shown here, since as a result of the printing process, differences between the original and balanced pair are not too visible. When viewed on a monitor, however, the differences are clear, and for the balanced pair, homologous points are seen to have closer matching characteristics. The reader is referred to a web page for the results of the balance algorithm, where the first field of both the original and balanced images are shown for both test sequences:
`<http://www.inrs-telecom.uquebec.ca/users/viscom/publications/>`.

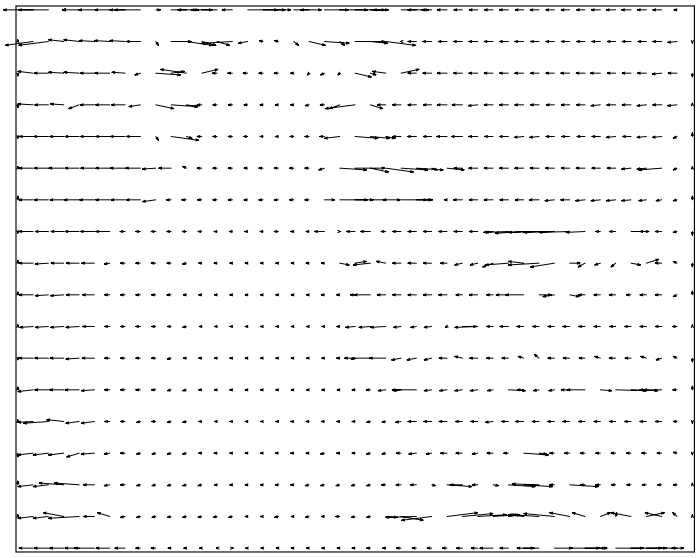
Disparity field estimations based on the balanced stereo pair are significantly improved. Figure 4.6 compares disparity fields estimated for *piano* using 2-D exhaustive search BM from (a) the original images and (b) the balanced images.

The estimated vector fields for both *flower* and *piano* have many fewer ambiguous matches than those found in Figure 4.5. For example, the piano player's head has become much more defined and the section of the piano itself which before had so many spurious matches, is now more accurately estimated. Problems in other regions remain, but will be tackled in due course. The gains in estimation are greater for test sequence *piano* which suffers from greater mismatches between the left and right images.

The incremental improvements that follow are all estimated from luminance- and chrominance-balanced stereo pairs.



(a) Estimation from original images



(b) Estimation from balanced images

**Fig. 4.6** Comparison of estimated disparity fields using 2-D exhaustive-search BM based on (a) original and (b) balanced images.

### 4.3.4 Luminance- & chrominance-based disparity estimation

Until now, disparity estimation has been carried out based on the luminance components of the left and right images only. When trying to obtain correspondence basing the match on luminance only, problems arise in areas where luminance detail is low. One may be able to obtain a more robust estimate in these areas by forcing a luminance match as well as a *colour* match.

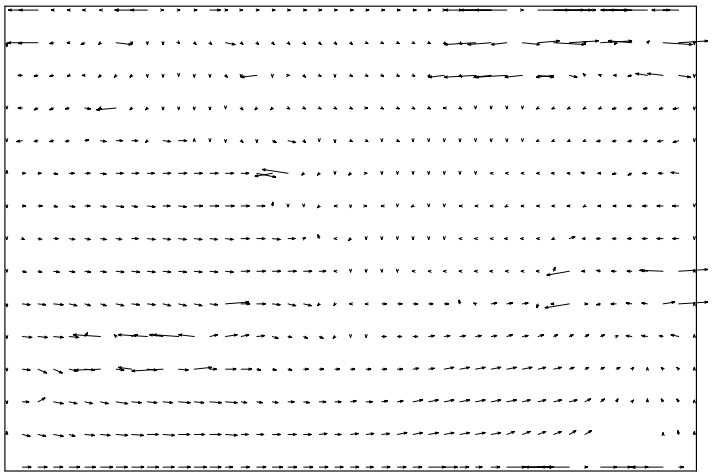
To implement a three-component (Y-U-V) match, the minimization in (4.4) is modified to accommodate the additional colour terms. For simplicity of notation, we use $(m, n, \varepsilon\, \underline{d}_{ij})$ to mean $(m + \varepsilon\, d_{ij_x}, n + \varepsilon\, d_{ij_y})$, where $\varepsilon = \alpha$ or $-(1-\alpha)$. Then, the minimization is:

$$\begin{aligned}\hat{\underline{d}}_{ij} = \arg\ \min_{\underline{d}_{ij}} \Bigg( & \gamma \cdot \sum_{(m,n)\in\beta_{ij}} \left| I_L^Y(m, n, \alpha\, \underline{d}_{ij}) - I_R^Y(m, n, -(1-\alpha)\, \underline{d}_{ij}) \right| \\ & \delta \cdot \sum_{(m,n)\in\beta_{ij}} \left| I_L^U(m, n, \alpha\, \underline{d}_{ij}) - I_R^U(m, n, -(1-\alpha)\, \underline{d}_{ij}) \right| \\ & \epsilon \cdot \sum_{(m,n)\in\beta_{ij}} \left| I_L^V(m, n, \alpha\, \underline{d}_{ij}) - I_R^V(m, n, -(1-\alpha)\, \underline{d}_{ij}) \right| \Bigg).\end{aligned} \tag{4.10}$$

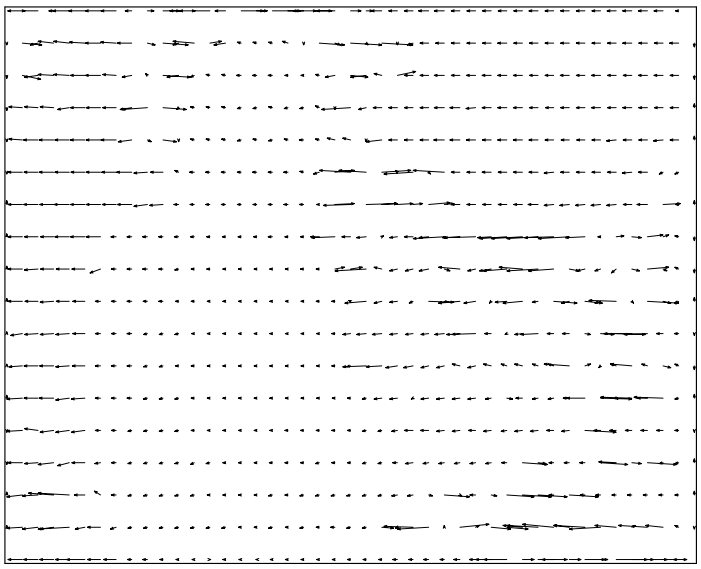
The individual image components are denoted by superscripts $''Y''$,$''U''$, and $''V''$. The relative contribution of each image component to the disparity estimation is controlled by the weights $\gamma, \delta$ and $\epsilon$. With the constraint $\gamma + \delta + \epsilon = 1$, one option is to give equal weight to each component, i.e., $\gamma = \delta = \epsilon = \frac{1}{3}$.

In regions of low luminance detail, the chrominance components could offer distinguishing features which will permit a more accurate match. In regions of low colour detail, it is the luminance information which potentially supplies distinguishing object characteristics. In this way, the three image components complement each other and work to obtain a more accurate estimation. It is also beneficial that the algorithm forces a colour match on input test sequences which have now been colour-balanced as well (Section 4.3.3).

Figure 4.7 shows disparity estimation results using the proposed luminance- / chrominance-based approach to BM for both test sequences. The estimated vector fields are more accurate using this technique than those found in Figure 4.6. The piano



(a) *flower*



(b) *piano*

**Fig. 4.7** Estimated disparity fields using 2-D exhaustive search luminance-/chrominance-based block matching.

player's head in (b), for example, has become even more identifiable from the disparity field. Since there is low luminance detail in this area, the chrominance components supply additional information used to distinguish the head from the background. Admittedly, visible differences in vector fields are subtle. In Chapter 5, when actual reconstructions are examined, the benefits of this method will be more obvious.

### 4.3.5 Spatial smoothness constraint

Previous sections tackled the problem of ambiguous matches in low-textured areas by trying to identify the cause of these incorrect matches. Although they are successful, a small but important percentage of vectors in Figure 4.7 remain inaccurate.

This section attempts to further correct ambiguous vectors by allowing correct matches to influence incorrect matches. For example, in the top right corner of the estimated vector field for sequence *flower* (see Figure 4.7(a)), one notices a number of large disparity vectors among a majority of small ones. The actual horizontal parallax associated with this region of the image (the brick wall in the background) is about 2 pixels. This is consistent with *most* vectors in the region.

Since the overall vector fields obtained thus far are mostly accurate, reliable vectors can be propagated into low-textured areas where vectors are typically unreliable. Mathematically, this is done through *regularization*, e.g., a smoothness constraint. Regularization, as discussed in detail in the context of pixel-based approaches in Section 3.4.2, penalizes disparity vectors which are very different from their neighbours. This forces a local likeness between adjacent vectors, and could eliminate problems like the ones in *flower* discussed above.

However, one must be careful not to impose too much regularization, since adjacent pixels belonging to objects from different depth planes should not be forced to have similar disparity vectors. In our case, only a small degree of smoothing is required, since estimated vector fields are accurate for the most part. However, smoothing across boundaries is inherent in the regularization approach presented here, and some degree of distortion is expected near object boundaries.

The minimization is modified to accommodate the regularization (smoothness)

term to give

$$\hat{\underline{d}}_{ij} = \arg \min_{\underline{d}_{ij}} \big( U_m(\underline{d}_{ij}) + \lambda \cdot U_s(\underline{d}_{ij}) \big), \tag{4.11}$$

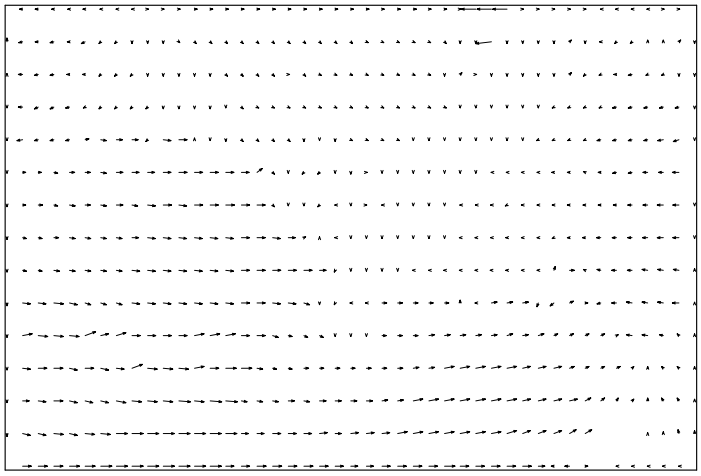
where $U_m(\underline{d}_{ij})$ is the matching error between blocks in the left and right images defined in (4.10), and $U_s(\underline{d}_{ij})$ is a sum of absolute-value differences between $\underline{d}_{ij}$ and its second-order neighbours. The parameter $\lambda$ controls the compromise between the closeness of the solution to the original data and the degree of smoothness. Given a candidate disparity vector $\underline{d}_{ij}$, $U_s(\underline{d}_{ij})$ is defined as

$$U_s(\underline{d}_{ij}) = \sum_{\substack{m=i-1,\\ m\neq i}}^{i+1} \sum_{\substack{n=j-1,\\ n\neq j}}^{j+1} \big( |d_{ij_x} - d_{mn_x}| + |d_{ij_y} - d_{mn_y}| \big). \tag{4.12}$$

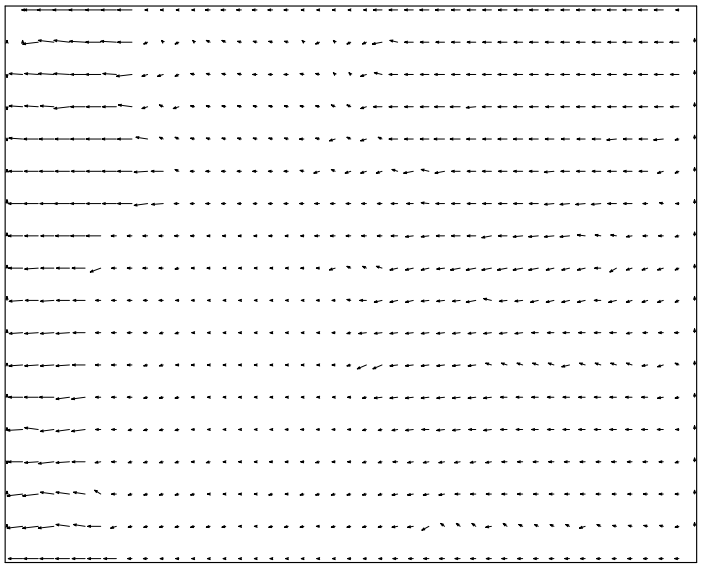
Computationally, the introduction of regularization into the minimization transforms the problem into an iterative algorithm; vectors influence each other because of the $U_s$ term, and a few iterations are required in order to reach convergence. This increases computational complexity of the algorithm. The stopping criterion for the algorithm is typically a fixed number of iterations, or maximum allowed energy (difficult to establish), or convergence rate (energy decrease over a fixed number of iterations). Here we use a fixed number of iterations.

Figure 4.8 shows the estimated vector fields for both test sequences using a smoothness constant of $\lambda = 15$. Both fields have become much more regular, and a significant number of ambiguous matches have been eliminated. Notably, the ambiguous matches in the top right corner of *flower* have been corrected. In addition, one can more easily trace out image objects or structures, such as the large flowerpot in the foreground of the image.

However, note that the disparity field for test sequence *piano* has perhaps been *over*-smoothed. Although estimated vectors for the piano player's body and head are now very accurate (there are no ambiguous matches in either), vectors surrounding the player's head have been influenced. One can see the effect of smoothing in this region, where vectors have become small. As a result, it is now more difficult to distinguish the boundary of the player's head from the vector field since this region



(a) *flower*



(b) *piano*

**Fig. 4.8** Estimated disparity fields from 2-D exhaustive search luminance-/chrominance-based block matching with regularization ($\lambda = 15$).

has now been diffused. Nevertheless, the application of a smoothness constraint on the data offers a definite improvement, as we will see in Chapter 5, since distortions in low-textured regions (e.g., the blue background) are much more tolerable to the viewer.

### 4.3.6 Robust estimation

Until now, the matching term, $U_m$, in the minimization (4.11) has been arbitrarily set to the absolute value function. This section looks at the characteristics of two other *estimator* functions in order to determine whether the solution to the correspondence problem could be further improved by using a more robust cost function than the absolute value function.

Estimator functions may be characterized by their robustness to *outliers*. Consider the scenario where a block is partially covering the boundary between the piano player's head and the background. If a disparity vector is estimated for this block which favours the head, the background pixels are considered "outliers". In this case, the region within the block made up of pixels from the head dominates the search for an optimal disparity vector, and we will call it the "main-area" of the block. On the other hand, if a vector is chosen such that the background is well-matched and not the head, then pixels within the block which are part of the head are considered outliers, and the background is considered to be the main-area. The situation we want to avoid is the case where neither region of the block is well-matched, which makes it difficult to say which pixels are the outliers, and which are part of the main-area. This results in a poor match for the majority of pixels in the block. This situation can be avoided, to a large extent, by employing a robust estimator.

A *robust* cost function is one which can tolerate a high percentage of outliers, while still obtaining a good estimate for the main-area of the block; it is more immune to the bias of "bad" samples. The main-area of a block is typically comprised of pixels belonging to that object which covers the largest portion of the block. This is not always the case, however, since it depends on the robustness of the estimator and on the texture of the contained objects.

A means of quantifying the robustness of an estimator is by examining its *breakdown point*. The breakdown point is defined as the percentage of outliers tolerated

before the estimator "breaks down" and obtains an arbitrarily bad estimate for the main-area of the block. The maximum attainable breakdown point is 50% (see Theorem 4 in Chapter 3 of [29] for a proof). Intuitively, this makes sense since above 50%, it becomes impossible to distinguish between the "good" and "bad" pixels of a block.

For the purpose of high-quality disparity estimation, an estimator with a high breakdown point is required. The higher the breakdown point, the higher the degree of certainty that the majority of pixels in any given block will be well-matched. Furthermore, since a robust cost function helps avoid the situation where all regions in a block are poorly-matched, such a problematic block, as it has been defined, will have significant differences in the number of outliers in each of its four *sub*-blocks. As we will see later in Section 4.3.7, this fact can be used to flag problematic blocks that fall on object boundaries and require careful attention.

The following discussion examines the robustness of three candidate cost functions for the difference term, $U_m$, and explains the motivation behind the selection of the most suited of these to the task of robust block matching.

### Quadratic function

One function often used for the matching term, $U_m$, and already discussed in the context of least-squares estimation, is the well-known quadratic function, shown in Figure 4.9.

As is eloquently discussed in [29], it only takes a single outlier to cause the least-squares (LS) estimator to break down and give an arbitrarily bad estimate for the majority of pixels in the block. Consider a block with very low texture which contains one single outlier pixel. It is possible that the cost associated with incorrectly matching this single pixel is greater than that associated with incorrectly matching the entire low-texture region. The optimal disparity vector for this block is that which favours matching of the single outlier-pixel. In this case, the declaration of the main-area of the block has been seriously biased by one single pixel, and the majority of the pixels in the block will suffer a poor match. Hence, since the quadratic function cannot tolerate any arbitrarily bad samples, it has a breakdown point of 0%.

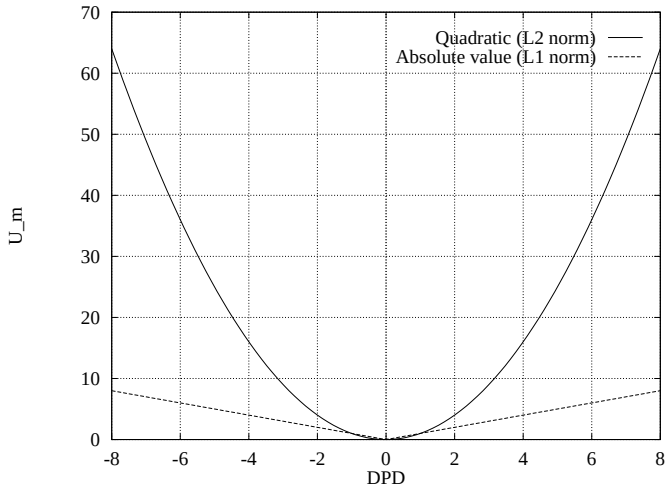



**Fig. 4.9** $L_1$ and $L_2$ estimators.

### Absolute value function

The next obvious step is to replace the quadratic (or $L_2$ norm) with the absolute value (or $L_1$ norm). From Figure 4.9, it is clear that outlying points are weighted less heavily by the $L_1$ norm, resulting in a reduced sensitivity to outliers. However, the estimator is not robust in that it still has a breakdown point of 0%. It only takes one very bad sample to skew the estimate to an arbitrarily bad solution[5].

### Geman-McLure function

To analyze the behaviour of an estimator, we define what is called an *influence function*. Denoting the estimator function by $\rho(\cdot)$, the influence function $\psi(\cdot)$ is defined as the derivative of $\rho(\cdot)$, i.e., $\psi(\cdot) = \dot{\rho}(\cdot)$. $\psi(\cdot)$ characterizes the weight a particular measurement has on the solution [30]. In [31], the solution to the gradient-based minimization process is seen to produce a weight for each measurement given by $\frac{\psi(x)}{x}$, which tends to zero for increasing x. In general, for the quadratic function, $\rho(x) = x^2$, and $\psi(x) = 2x$, hence the influence of outliers increases linearly and without bound. For the absolute value function, $\psi(x) = sign(x)$, hence the influence of outliers is still felt.

[5]See page 11 of [29] for experimental proof of the 0% breakdown point of the *least absolute value* estimator (the $L_1$ norm).

To increase robustness, *redescending* estimators are considered, where the influence of outliers tends to zero. One such function is the *Geman-McLure* (GM) estimator, $\rho(x;\sigma)$, used in the context of robust estimation in [31]. It is given by

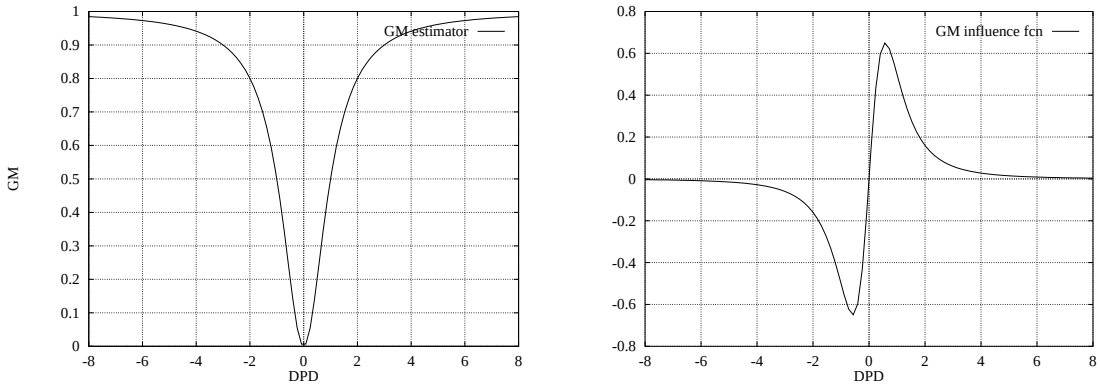
$$\rho_{GM}(x;\sigma) = \frac{\frac{x^2}{\sigma^2}}{1+\frac{x^2}{\sigma^2}}, \qquad \psi_{GM}(x;\sigma) = \frac{2x\sigma^2}{(\sigma^2+x^2)^2}, \tag{4.13}$$

where $\sigma$ is a scale factor. In our case, $x$ is the displaced pixel difference (DPD) between pixels forming a candidate pair in the left and right images, i.e.,

$$DPD = \left| I_L(i,j,\alpha\underline{d}_{ij}) - I_R(i,j,-(1-\alpha)\underline{d}_{ij}) \right|, \tag{4.14}$$

where for simplicity, we again use the notation $(m,n,\varepsilon\,\underline{d}_{ij})$ to mean $(m+\varepsilon\,d_{ij_x}, n+\varepsilon\,d_{ij_y})$.

Plots for both $\rho_{GM}(x;\sigma)$ and $\psi_{GM}(x;\sigma)$ are given in Figure 4.10 for $\sigma = 1$.




(a) GM estimator, $\rho_{GM}(\cdot)$ (b) GM influence function, $\psi_{GM}(\cdot)$

**Fig. 4.10** The Geman-McLure (GM) redescending estimator of (4.13) for $\sigma = 1$.

From Figure 4.10(a), one can see that outliers are not nearly as heavily weighted as with the $L_1$ or $L_2$ norms. A DPD of 6 has a cost of 36 with the $L_2$ norm, 6 with the $L_1$ norm, and 0.97 with the GM estimator. In addition, from Figure 4.10(b), the influence of outliers is seen to tend to zero.

The breakdown point of the GM estimator (4.13) is largely determined by the selection of an appropriate scale factor, $\sigma$. A robust method for automatic selection of the scale factor based on the pixel-samples is developed. Adopting the approach outlined in [31] and originating from [29], an estimate for $\sigma$ is derived from the median value of the absolute residuals using

$$\sigma_{ij} = 1.4826 \cdot median\,|r_{ij}|. \tag{4.15}$$

The constant in (4.15) is based on the fact that the median value of the absolute values of a "large enough" sample of unit-variance normal distributed one-dimensional values is $\frac{1}{1.4826}$ [31]. The residuals are given by $r_{ij}$, which is the set of DPD values between the corresponding blocks in the left and right images defined by $\underline{\hat{d}}_{ij}$. The number of samples over which the median is found is given by $N_x \times N_y$. The assumption is that, in the case of a typical block-size of $16 \times 16$, 256 samples is a "large-enough" sample. This median-based estimate for $\sigma$ offers excellent resistance to outliers, and in the limit, it can tolerate almost 50% of them [31].

To evaluate outlier-resistance of the automatic scale estimator in (4.15), it is tested for estimation on a single block of size 64x64. The block is cut out from test image *flower* at position (357,164) and shown in Figure 4.11. The Geman-McLure cost function with automatic scale estimation is used to estimate a single disparity vector for all pixels in this block, and for position $\alpha = 0.5$. The initial value for the scale constant $\sigma_0$, is calculated from (4.15) based on a zero-value disparity vector. Iteration 0 of the BM minimization is performed using the GM function with scale $\sigma_0$, and a disparity vector $\underline{\hat{d}}_0$, is obtained. For the next iteration (4.15) is once again used to obtain $\sigma_1$ based on the residuals from $\underline{\hat{d}}_0$, and in turn, $\sigma_1$ is used to compute $\underline{\hat{d}}_1$. The resulting DPD distributions for each pixel in the 64x64 block and for both iterations are shown in Figure 4.12 in the form of 3-D surface plots. The $x$- and $y$-axes define the pixel position within the block, and the $z$-axis is the corresponding DPD value. Notice the drastic decrease in DPD after one iteration due to the estimation of $\underline{\hat{d}}_1$ which is a more accurate disparity vector than $\underline{\hat{d}}_0$ for the majority of pixels in the block.

In order to compute the median value of the distributions in Figure 4.12, they are first numerically sorted to form an ordered sequence of numbers. To give an idea of

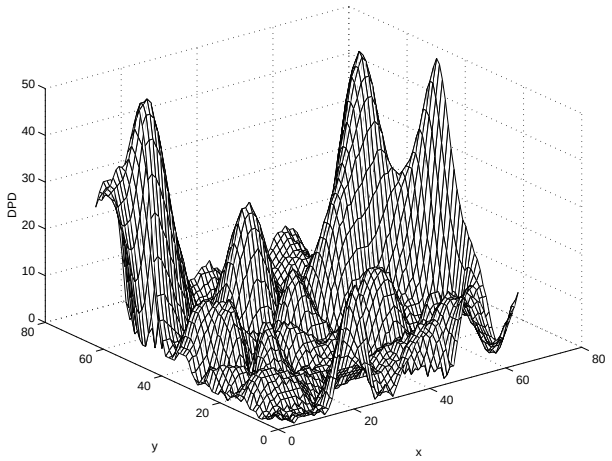
(a) left (b) right

**Fig. 4.11** Enlarged sub-window cut out from *flower* at position (357,164). The 64x64 block in the center of the sub-window is surrounded by 16 additional pixels on the left and right, and 2 above and below. This defines the state space for exhaustive-search BM.

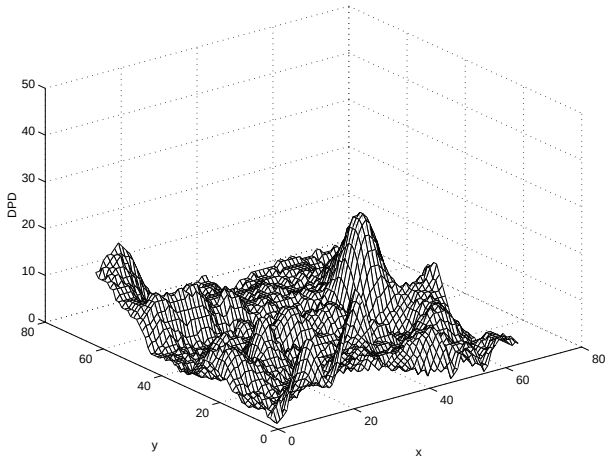
the change in the DPD over the whole block after one iteration, the sorted sequences of numbers for both iterations are subtracted and the resulting curve is shown in Figure 4.13. The $x$-axis defines the position in the ordered sequence, and the $y$-axis specifies the corresponding DPD value. The curve is very flat up to a sorted DPD pixel position of about 3000, meaning that the two distributions are very similar up to this point. After this, they begin to differ substantially. The median value is computed as the average of the values at positions 2047 and 2048 since there are 4096 measurements for this particular block size. Clearly the heavy activity at positions 3000 and higher do not affect the median value, and hence the scale estimation equation (4.15) is seen to be resistant to outliers.

In this particular case, the median value has decreased[6], resulting in a smaller scale constant for iteration 1 ($\sigma_1 < \sigma_0$). The smaller the scale constant, the narrower the Geman-McLure curve in Figure 4.10(a), and the closer the estimated vector gets to the correct disparity for the main-area of the block. A hierarchical approach is used in a coarse-to-fine fashion for this estimation (see Section 4.5), and the resulting disparity

[6]The curve in Figure 4.13 is obtained by subtracting the sorted DPD distribution at iteration 1 from the sorted DPD distribution at iteration 0.




(a) Iteration 0




(b) Iteration 1

**Fig. 4.12** DPD distributions from robust estimation of the image region shown in Figure 4.11 from (a) $\underline{\hat{d}}_0$ and (b) $\underline{\hat{d}}_1$.

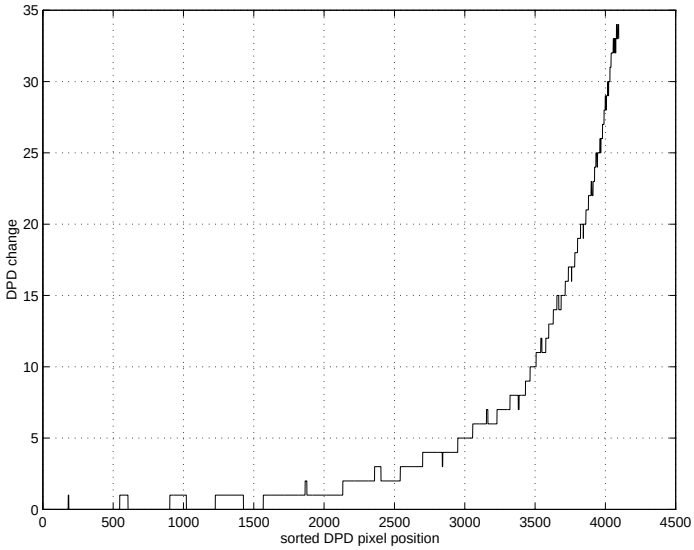


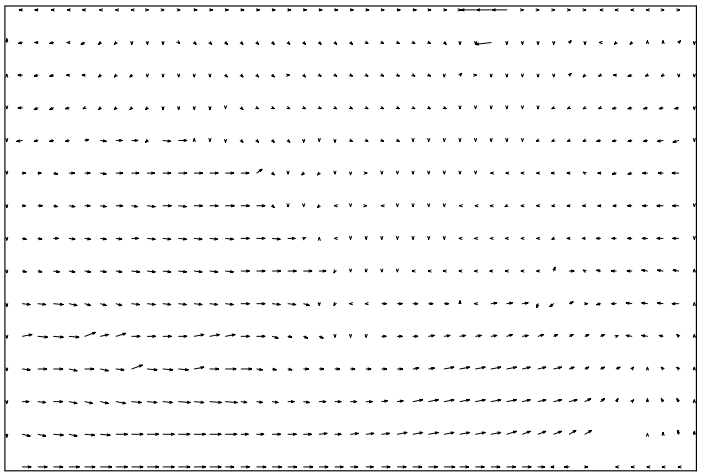

**Fig. 4.13** DPD vs. sorted pixel position. The difference in the sorted DPD distributions of Figure 4.12.

vector obtained after four iterations for the block in Figure 4.11 is $\hat{\underline{d}} = (6, -\frac{1}{2})$, which is exactly true for the main-area of this block.

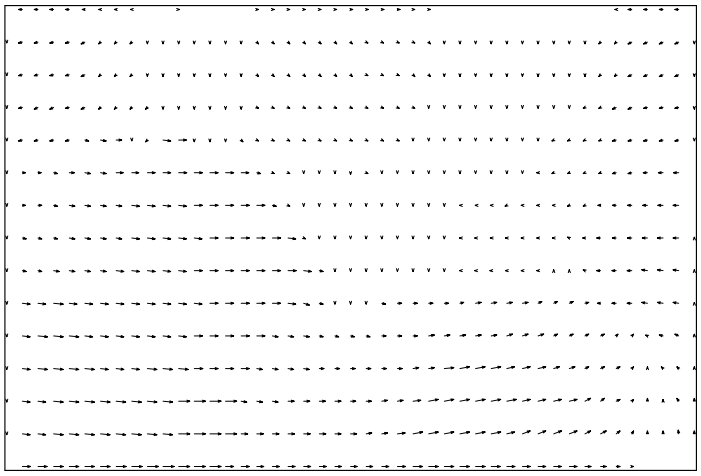
From this analysis, it is clear that one could obtain good resistance to outliers by using the GM function with automatic scale estimation. Incorporating this cost function as the difference term, $U_m$, in the minimization is straightforward. The calculation for $\sigma$ given in (4.15) is done for each block $(i, j)$ in the intermediate image, and for each iteration[7]. Note that since the maximum penalty assigned to any one sample is much lower with the Geman-McLure function than that of the previously used $L_1$ norm, the smoothness constant, $\lambda$, is lowered so that the relative contributions from the difference and smoothness terms remain about the same. For experimentation, it is set to $\lambda = 2.5$.

Due of the increased robustness towards outliers offered by the Geman-McLure function, the estimated disparity fields are more accurate than before. A comparison between disparity fields, one obtained using the $L_1$ norm, and the other using the GM estimator, is shown for both test sequences in Figures 4.14 and 4.15.

[7]Remember that the algorithm became iterative ever since the introduction of regularization (smoothness).

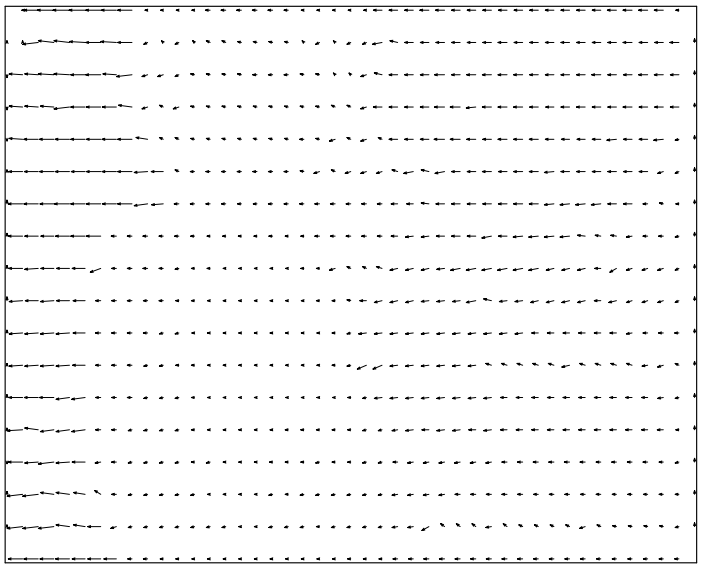


(a) absolute value estimator

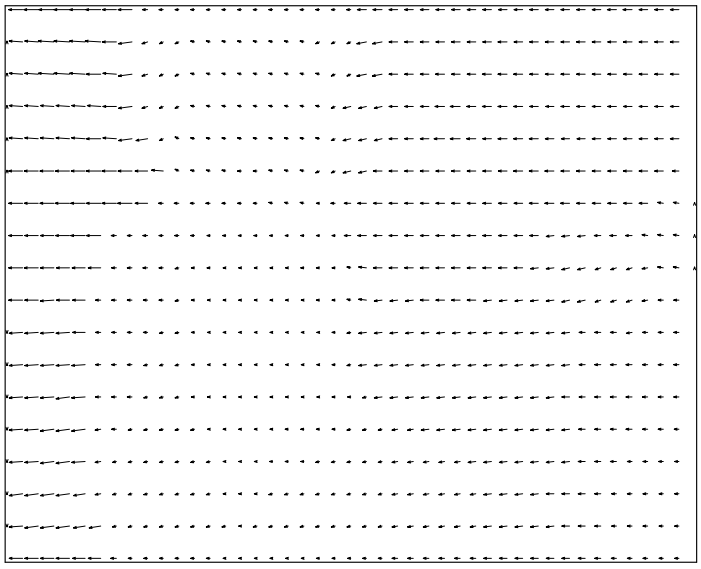


(b) Geman-McLure estimator

**Fig. 4.14** Comparison of estimated disparity fields between MAD and robust estimation using the Geman-McLure estimator for test sequence *flower*.



(a) absolute value estimator



(b) Geman-McLure estimator

**Fig. 4.15** Comparison of estimated disparity fields between MAD and robust estimation using the Geman-McLure estimator for test sequence *piano*.

Although differences are subtle between the disparity fields, there is one excellent example which confirms the increased robustness of the GM function. In the original *flower* image, there are three letters, "NHK", in the bottom right corner of the image (refer to the original image on page 47). Although these letters are evidently not part of the original, natural scene, the pixels that form them do nevertheless have an impact on the estimation in this area. For the disparity field found from MAD estimation shown in Figure 4.14(a), notice that the region corresponding to these three letters is estimated as having zero disparity (null-vectors). Since the letters are placed at exactly the same location in the left and right images, and hence have zero parallax, this is correct. Therefore, according to the absolute value estimator, the letters were declared as the main-area of blocks falling in this region and have hence dominated the search.

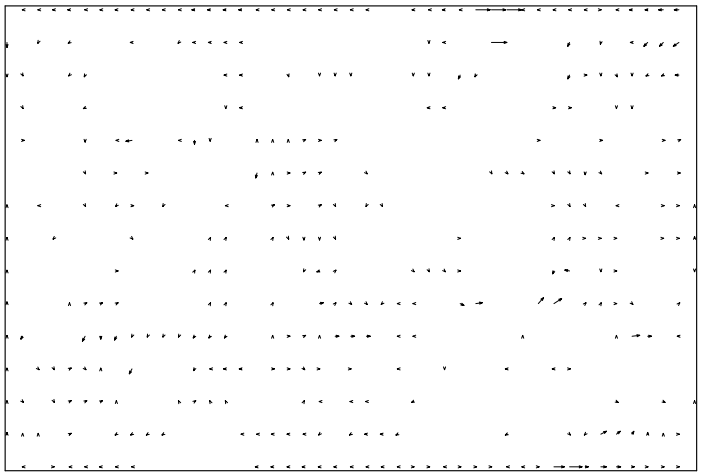
In contrast, notice how vectors in this same region in the disparity field found using the Geman-McLure estimator, shown in Figure 4.14(b), are not zero. In fact, they are more like their neighbouring vectors. In this case, for blocks falling in this region, the GM estimator has declared the background pixels as the main-area, and pixels belonging to the "NHK" letters as outliers. This scenario is closer to what we would expect since the background pixels occupy a greater portion of the blocks in this region. The GM estimator has not allowed the bright pixels from the "NHK" to dominate the search and incorrectly bias the estimate. This is exactly the kind of advantages a robust estimator can offer.

To get a better idea of exactly where the robust estimator has made a difference in the estimation, consider the *difference* fields shown in Figure 4.16 for both test sequences. Difference fields are simply a point-by-point subtraction of each disparity vector from two input fields.

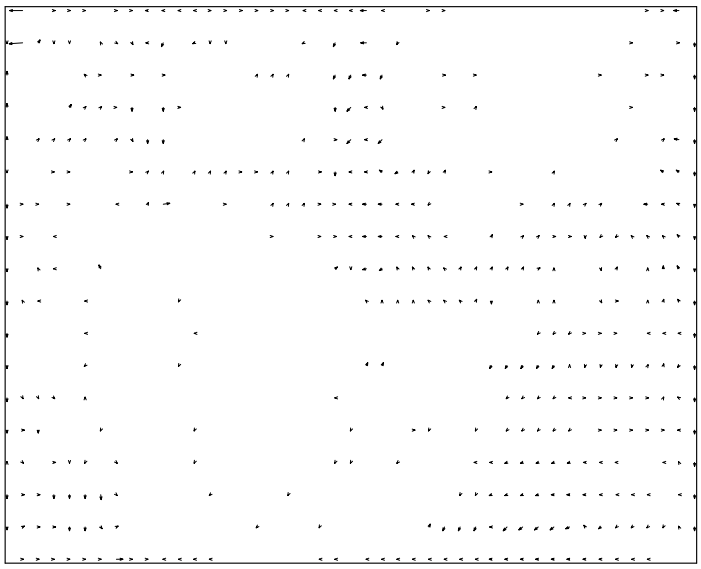
### 4.3.7 Quadtree structure

Several improvements to block matching have been proposed thus far, and the current estimated disparity fields shown in Figure 4.14 are by now very accurate. As will be shown in Chapter 5, one obtains high quality intermediate view reconstructions based on these vector fields.

Problems that remain, however, are imprecise estimations near object boundaries.



(a) *flower*



(b) *piano*

**Fig. 4.16** Difference disparity fields between MAD and robust estimation using the Geman-McLure estimator. Reference fields are shown in Figure 4.14.

The current model is incapable of supporting depth and disparity discontinuities since all pixels in a block are described by a single disparity vector. To explain, consider two objects, object 1 and object 2 in Figure 4.17, which have different parallax values; object 1, 1 pixel, object 2, 10 pixels. Object 2 covers object 1, as shown. Block $(i, j)$ is such that it partially covers the boundary between these two objects. Since the model will assign only one vector to each pixel in this block, there is a problem. The support of the model needs to be adjusted if this object boundary is to be well reconstructed.

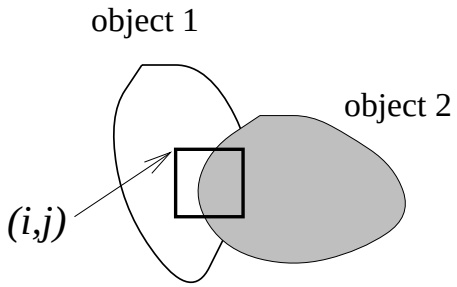



**Fig. 4.17** Scenario demonstrating weakness of current model at reconstructing boundaries. Block $(i, j)$, which is described by a single disparity vector, is shown to cover the boundary between objects 1 and 2, which have require different disparity vectors.

In order to more accurately represent a greater number of pixels in such a block, it is split into four, equal-size sub-blocks. The sub-blocks will have a greater probability of not covering any depth discontinuities, and are reestimated. A more accurate description of member pixels is expected since four disparity vectors are now used to describe the pixels of the original block. Sub-blocks that still fall on object boundaries can be further split into four. In so doing, one ends up with a situation similar to that in Figure 4.18, which shows three levels of splitting. The smallest blocks are the ones closest to the boundary. This scenario is called the *quadtree structure*.

### Automatic detection of problematic blocks

The quadtree structure could theoretically be applied to every block in the intermediate image. However, this would put all previous efforts at obtaining reliable disparity

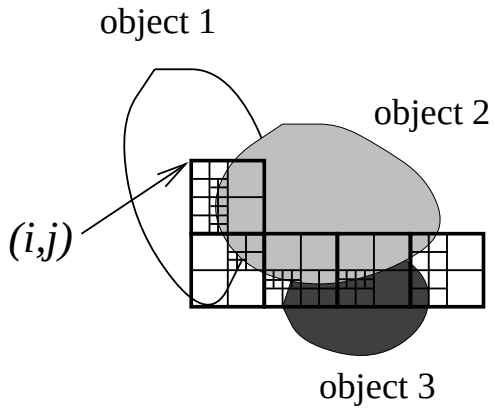



**Fig. 4.18** Block-splitting based on the quadtree structure. Problematic blocks are recursively split until all blocks no longer overlap object boundaries.

estimates to waste. Why split a block if one vector is sufficient at describing its member pixels? On the contrary, for reasons of simplicity and of reduced bandwidth in a video telecommunications system, transmitting one vector per block is beneficial and one should try to do this wherever possible[8]. That said, the issue now is to find a robust way of targeting blocks that cover depth discontinuities in the image, and only split and reestimate those *problematic* blocks.

One way of selecting problematic blocks is to apply thresholding to the difference, or matching-error term in the minimization, $U_m(\cdot)$ (first introduced in (4.11)). In this case, for any pair of blocks that gives a difference energy greater than a specified threshold, the corresponding block in the (intermediate) image is split and reestimated. Although simple, this method has proven to be not very robust, since results are highly image- and threshold-dependent.

Alternatively, given a disparity vector which represents a block in the intermediate image, differences in certain characteristics of the independent sub-blocks could be exploited. Consider once again block $(i, j)$ in Figure 4.17, which is known to be problematic. Since there are a greater number of pixels in the block that belong to object 2, there is a good chance this region will dominate the search and hence

[8]In reality, in the context of a transmission system, the quadtree structure does complicate things slightly since the support of each vector needs to be known at the receiver. Although inexpensive since it is regular, the map representing the quadtree structure needs to be transmitted.

would be declared the main-area of the block[9]. Object 1 is defined to be associated with a horizontal parallax of 1 pixel, and object 2, 10 pixels. If a disparity vector of magnitude 10 is found to be optimal, then block $(i, j)$ will have a large number of outliers in two of its *sub*-blocks (the two on the left, which are mostly made up of pixels from object 1), and few, if not none, in the other two *sub*-blocks (the two on the right), as shown in Figure 4.19.

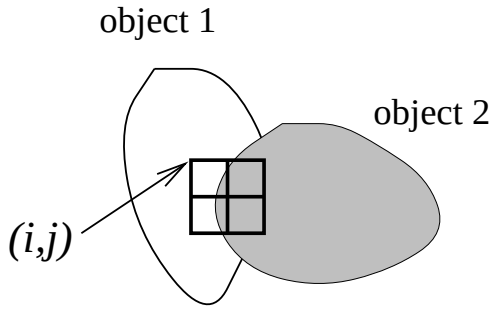



**Fig. 4.19** Estimated disparity vector for problematic block $(i, j)$ favours object 2. Sub-blocks on the left, which for the most part contain pixels from object 1, have many outliers. Sub-blocks on the right have very few outliers.

In this example, an alarm sounds for block $(i, j)$ due to the fact that its four sub-blocks have such a discrepant number of outliers. Depending on the criteria used to declare that the relative number of outliers in the four sub-blocks is discrepant, a block such as this one would be selected for splitting. These sub-blocks would then be independently reestimated, and a better representation of member-pixels obtained.

It is a side-effect of using a *robust* estimator that one can be relatively certain that a problematic block will have a significantly different number of outliers in its four sub-blocks. It is this side-effect that is exploited in the proposed approach towards automatic detection of problematic blocks.

Before describing the approach in detail, we need to mathematically define an "outlier" so that we may appropriately tag pixels in a block. One notices from the plot in Figure 4.10 that the GM curve begins to taper off at a DPD, or difference

[9]Depending on the texture of the different objects defining the boundary, it is not necessarily the object which occupies the largest portion of the block that will dominate the search, but rather the area which, if well-matched, results in the lowest overall cost.

value, of about $2\,\sigma$ (or 2 in the figure, for $\sigma = 1$). This corresponds to an error of 0.8. Roughly, this means that at a difference value greater than $2\,\sigma$, the higher the DPD, the more the sample is treated as an outlier since the error it contributes begins to saturate to 1.0. At DPD values of less than $2\,\sigma$, all samples are equally penalized since the curve is roughly linear there. Therefore, it seems appropriate that an outlier is declared for any DPD value of $2\,\sigma$ or above.

The algorithm for automatic detection of problematic blocks starts with a disparity field estimated using the current approach. Then, for each block $(i, j)$ in the intermediate image, the following steps are carried out:

1. compute the number of outliers for each of the four sub-blocks,
2. compute the average DPD over all pixels in the block,
3. execute the decision-making process outlined in Figure 4.20.

The flow chart in Figure 4.20 represents a three-step decision-making process to determine whether a block requires splitting. The three steps are denoted by the questions, $Q1$, $Q2$ and $Q3$. Together, they permit a reasonable compromise between the number of blocks declared as problematic (or unreliable), and the number of blocks declared as reliable.

**Q1** Blocks with a very good estimate will have a low average DPD for all pixels in the block. Such blocks will not pass $Q1$, and hence will not be split. The problematic blocks that we are after would not have a low average DPD.

**Q2** If the total number of outliers in a block is small, it means that most pixels in the block were declared as the main-area, and hence there is a very good chance such a block does not cover a depth plane discontinuity. Such blocks would not pass $Q2$, and hence are not split.

**Q3** Finally, for all four sub-blocks, if the ratio of the maximum to the minimum number of outliers in the four sub-blocks is high, there is a good chance that such a block *does* cover an object boundary, hence requires splitting. If not, then we are dealing with a block which has been poorly estimated throughout, has many outliers,

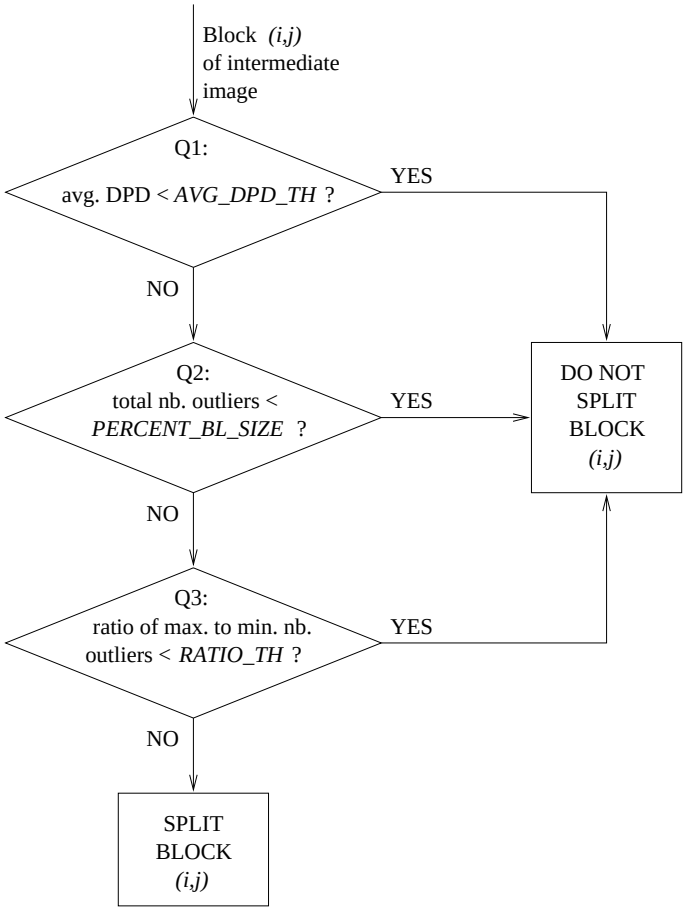



**Fig. 4.20** Decision flow chart for automatic detection of problematic blocks. Performed for each block $(i, j)$ of the intermediate image.

and thus reestimation will not help. This scenario typically occurs only in areas of very high texture, and is rare.

Decision-making steps 1, 2 and 3 are all associated with certain threshold values. Table 4.3 lists the significance of these, as well as their default values used in experimentation.

The algorithm for automatic detection of problematic blocks is applied to test sequences *flower* and *piano*. The results of which blocks are declared as unreliable are shown in Figure 4.21. Chapter 5 will give a more detailed discussion of the results, where the significance of the current image reconstructions shown in Figure 4.21 will

| Step | Symbol name | Significance | Default value |
|---|---|---|---|
| $Q1$ | *AVG_DPD_TH* | Average value of DPD, below which a block is declared reliable. | 5.0 |
| $Q2$ | *PERCENT_BL_SIZE* | Percentage of block size representing tolerable number of outliers below which a block is declared reliable. | 10% |
| $Q3$ | *RATIO_TH* | Ratio of maximum to minimum number of outliers in the four sub-blocks of a block, above which a block is declared unreliable. | 2.0 |

Table 4.3 Threshold values of the decision-making process for automatic detection of problematic blocks.

be more clear. There will also be a discussion on the relative (in)sensitivity of the algorithm to the chosen threshold values of Table 4.3.

### Recursive estimation of problematic blocks

Now armed with a robust estimator and dependable problematic block detector, the task is to reestimate sub-blocks of targeted blocks. Since some selected blocks in Figure 4.21 do not actually cover a depth discontinuity, the hope is that the reestimation process is robust enough not to worsen the estimates in these areas.

The reestimation of sub-blocks is done as before, using 2-D, exhaustive search block matching based on components Y-U-V and using regularization and robust estimation. Of course, $N_x$ and $N_y$ are reduced by half for each level of splitting. Smoothing, however, is done only with neighbouring full-size blocks (i.e., blocks that have not been chosen for splitting). Mathematically, for the sub-block at position

(a) *flower*

(b) *piano*

**Fig. 4.21** Results of the proposed algorithm for automatic detection of problematic blocks. Threshold values used to declare a problematic block are listed in Table 4.3.

$(i, j)$, the smoothness term becomes

$$U_s(\underline{d}_{ij}) = \sum_{\substack{m=i-1,\\ m\neq i}}^{i+1} \sum_{\substack{n=j-1,\\ n\neq j}}^{j+1} \delta_{mn} \cdot \big( |d_{ij_x} - d_{mn_x}| + |d_{ij_y} - d_{mn_y}| \big), \tag{4.16}$$

where $\delta_{mn}$ is 1 if and only if the sub-block at $(m, n)$ belongs to a full-size block which was not chosen for splitting. This modification to the smoothness term is motivated by the fact that blocks which have been tagged as problematic are considered unreliable, and therefore their disparity vectors should not influence those of neighbouring blocks. Estimation is done based on luminance- and chrominance-balanced images, as usual.

Since neighbouring disparity vectors influence each other as a result of the smoothness constraint, it is important to first correct those problematic blocks that have the fewest number of unreliable neighbours. In this way, one recursively increases the number of reliable neighbours surrounding problematic blocks. The first step is hence to classify selected problematic blocks in terms of the number of unreliable *sub*-blocks surrounding them in a second-order neighbourhood. This results in classes of blocks; `class0` has 0 neighbouring sub-blocks which are unreliable, `class1` has 1, etc. . . , as shown in the example of Figure 4.22.

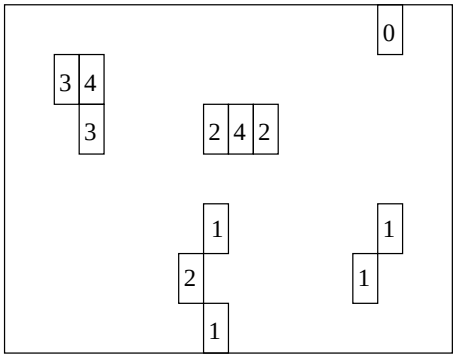



**Fig. 4.22** Example of a set of blocks declared as unreliable by the proposed algorithm for automatic detection of problematic blocks. A problematic block is classified according to the number of unreliable neighbouring *sub*-blocks it has. Full-size blocks are shown.

The procedure for reestimating the established classes can be written algorithmically as:

```
for i=0 to MAX_LEVEL
  for j=i downto 0
    while (!class_empty(j))
      reestimate_class(j);
      update_class(j);
    end
    update_all_classes();
  end
end
```

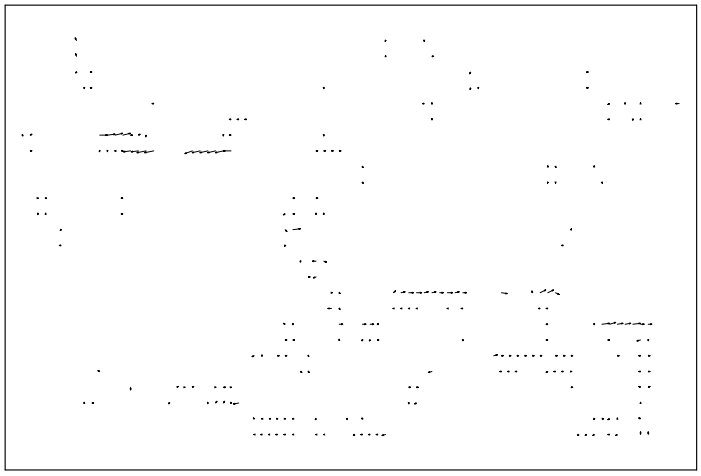
where `MAX_LEVEL` is the maximum number of unreliable sub-block neighbours a problematic block can have; `MAX_LEVEL`=12.

Once the four sub-blocks of a block are reestimated, a reclassification of neighbouring problematic blocks is required. For this, the `update_all_classes()` procedure is used. The structure of the pseudo-code is such that classes are tackled in the following order: `class0`, `class1`, `class0`, `class2`, `class1`, `class0`, `class3`, `class2`, `class1`, `class0`, etc... The `while` loop is used to ensure that all members of a class are dealt with before moving to the next one.

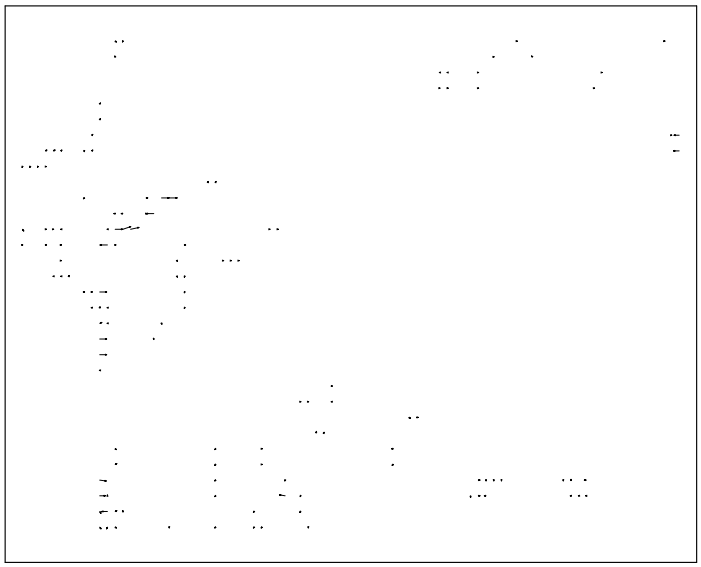
Resultant *difference* disparity fields for no splitting vs. one-level of splitting are shown in Figure 4.23 in the form of one vector per 8x8 block.

The interesting thing to note in the shown fields is that any substantial differences that exist between the disparity fields exist around object boundaries. Comparing the difference field for *flower* in Figure 4.23(a) vs. the blocks selected for splitting in Figure 4.21(a), notice that the estimations for all selected blocks which did not in fact cover any depth discontinuities did not change very much, if at all. This is excellent since it means the algorithm does not worsen estimates for blocks that do not really need splitting. However, notice that significant differences exist around the top boundary of the large flowerpot in the left half of the image, as well as near the boundary of the foreground flowers and the background counter around the lower half of the image, near the middle. Vectors in the difference fields are large in these areas.

For *piano*, the situation is similar. Looking at the difference field in Figure 4.23(b), again notice that the most significant changes in estimated vectors occur around the boundary of the left side of the player's back and the background. Estimates for all



(a) *flower*



(b) *piano*

**Fig. 4.23** Difference disparity fields between no splitting vs. one level of splitting.

selected blocks which fall within the shirt did not change much, if at all. This is exactly what we would hope since such blocks do not in fact require splitting. In Chapter 5, image reconstructions will show that the changes around these object boundaries are, for the most part, for the better. The quadtree approach, with its corresponding reestimation algorithm, offers considerable improvements in the quality of boundary reconstructions for intermediate pictures. In Chapter 5, we will look at the effect of further levels of quadtree splitting down to a 4x4 block size, and even 2x2.

## 4.4 Pixel-based disparity estimation

Alternatives to the block-based approaches to disparity estimation for intermediate view reconstruction are the pixel-based approaches. An approach based on regularization was presented in Section 3.4.2 in the context of disparity-compensated predictive coding. In this section, it is adapted to the problem of IVR.

### 4.4.1 Adaptation to intermediate view reconstruction

The mathematical equations in Section 3.4.2 describing the regularization approach must be modified in order for the estimated disparity fields to be used in the reconstruction of intermediate views. Recall the minimization in question,

$$\min_{u} \mathcal{E}(u) = \sum_{(i,j)\in I_L} \mathcal{P}(u) + \lambda \mathcal{R}(u). \tag{4.17}$$

The difference term $\mathcal{P}(u)$ is modified as follows to fit the model described in Section 4.2:

$$\mathcal{P}(u) = \big[I_L(i+\alpha u, j) - I_R(i-(1-\alpha)u, j)\big]^2, \tag{4.18}$$

and $\mathcal{R}(u)$ remains unchanged as in (3.7).

As before, minimization is carried out by equating the derivative of the cost function in (4.17) to zero and solving for $u(i,j)$, the disparity vector at the intermediate

picture point $(i, j)$. This gives

$$u = u^* + \frac{1}{\lambda}\big[I_L(i + \alpha u, j) - I_R(i - (1 - \alpha)u, j)\big]\big[\alpha I_{L_x} + (1 - \alpha)I_{R_x}\big], \tag{4.19}$$

where $(i, j)$ is the understood argument for all references made to $u$, and where $u^*$ is defined as the local average of the disparity vector at the point $(i, j)$. The main novelty in this equation is the weighted contributions from right *and* left image gradients as opposed to (3.11), where the gradient at the right picture point only appears.

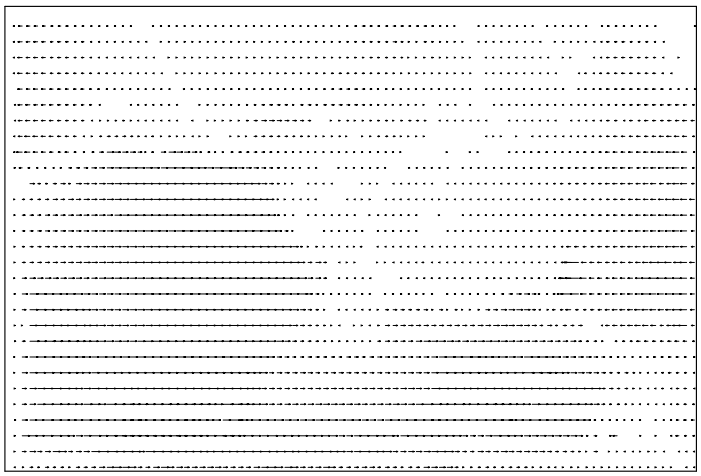
For $\alpha = 0.5$ and $\lambda = 500$, the estimated disparity fields for *flower* and *piano* are shown in Figure 4.24. Estimations are done on the luma- and chroma-balanced images, as always. For optimal printout clarity, they are presented in the form of one vector per 8x8 block of pixels, as opposed to one vector per 16x16 block of pixels for the block-based results.

The regularity of the disparity fields is immediately noticeable, and no ambiguous matches are seen within objects such as the player's back in *piano*, or the large flowerpot in *flower*. The image reconstructions based on these will be examined in Chapter 5.

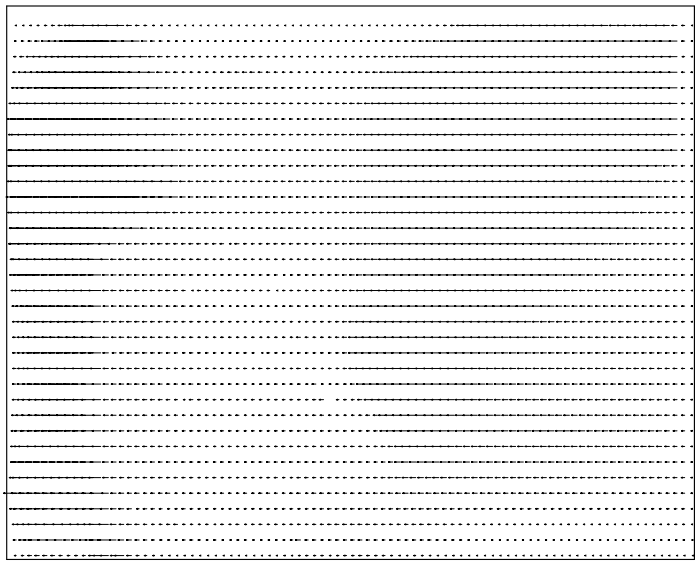
### 4.4.2 Vertical disparity

In practice, test sequences do not perfectly satisfy the assumption of parallel acquisition cameras. This section will show how the method could therefore be modified to allow for a two-dimensional disparity field, $(u, v)$. As a result, one obtains two iterative equations of the form (3.12). However, since we do not want to introduce large vertical components to the disparity vectors, a third constraint is applied in (4.17) to make sure the vertical disparity component is separately constrained. This results in the following minimization:

$$\min_{u,v} \mathcal{E}(u, v) = \sum_{(i,j)\in I_L} \mathcal{P}(u, v) + \lambda\mathcal{R}(u, v) + \gamma\mathcal{D}(v), \tag{4.20}$$



(a) *flower*



(b) *piano*

**Fig. 4.24** Estimated disparity fields using the pixel-based regularization approach for intermediate picture position $\alpha = 0.5$. Regularization constant is $\lambda = 500$.

where

$$\begin{aligned}
\mathcal{P}(u,v) =& \big[I_L(i+\alpha u, j+\alpha v) - I_R(i-(1-\alpha)u, j-(1-\alpha)v)\big]^2, \\
\mathcal{R}(u,v) =& \big[(u(i,j)-u(i,j-1))^2 + (u(i,j)-u(i,j+1))^2 + \\
& (u(i,j)-u(i-1,j))^2 + (u(i,j)-u(i+1,j))^2 + \\
& (v(i,j)-v(i,j-1))^2 + (v(i,j)-v(i,j+1))^2 + \\
& (v(i,j)-v(i-1,j))^2 + (v(i,j)-v(i+1,j))^2\big], \\
\mathcal{D}(v) =& v^2(i,j),
\end{aligned} \tag{4.21}$$

$\mathcal{R}(u,v)$ is modified to implement smoothing over a second order neighbourhood. It also permits regularization to be imposed on the vertical components of vectors as well.

As before, the minimization procedure yields an iterative equation for the $y$-component of disparity, $v$, in the form of

$$\begin{aligned}
v^{n+1} = \lambda v^{*n} + \frac{1}{\lambda+\gamma}\big[I_L(i+\alpha u, j+\alpha v) - I_R(i-(1-\alpha)u, j-(1-\alpha)v)\big] \times \\
\big[\alpha I_{L_y}(i+\alpha u, j+\alpha v) + \\
(1-\alpha) I_{R_y}(i-(1-\alpha)u, j-(1-\alpha)v)\big].
\end{aligned} \tag{4.22}$$

The inclusion of a vertical disparity component offers some improvements to the estimated disparity fields. With $\gamma = 1000$, vertical components are kept small, yet estimates are more accurate. We will see in Chapter 5 how image reconstructions are slightly improved as a result of this small modification.

## 4.5 Hierarchical approach or pyramidal

The hierarchical approach is not a disparity estimation technique, as such, but rather a means of applying a known estimation algorithm to a set of images forming a pyramid. The approach is typically known to offer improved estimation quality by avoiding local minima, and increased computational efficiency [20].

The pyramid is created through successive applications of filtering and subsam-

pling operations. Hence, starting from an original image size of 720x480 (level 0 – the highest resolution level), we create images of sizes 360x240, 180x120, and 90x60 (level 3 – the lowest resolution level), as depicted in Figure 4.25. Then, the chosen disparity estimation technique is applied to each level of the pyramid, starting with the lowest resolution level.

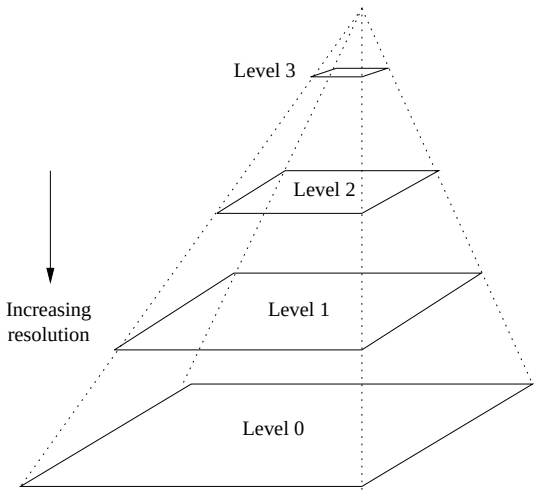



**Fig. 4.25** Hierarchical, multiresolution image representation.

The idea of the approach is that estimates at lower resolution levels serve to determine a rough estimate of the disparity. The estimate of the disparity vector at a lower level is passed onto the next higher level as an initial estimate. The higher resolution levels therefore serve to fine-tune the estimate.

In practice, the subsampling step may be skipped. Then the pyramid is made up of images that are all the same size, but successively more blurred as you move up the pyramid. In the context of hierarchical BM, this form for the pyramidal structure is beneficial. The low-pass filtering eliminates high frequency components in the image, allowing the exhaustive search approach to obtain gross estimates of the correct disparity at the filtered levels. The search window may be reduced as you move down the pyramid since we start from increasingly better initial estimates.

In the case of the pixel-based regularization approach, the subsampling operation is not skipped, forming a multiresolution pyramid. Since the initial estimate for disparity is the null field, the hierarchical approach permits the disparity estimator

to lock-on to large disparity vectors.

The hierarchical approach was implemented for both block- and pixel-based approaches examined herein. Its application was seen to reduce execution time by more than 50%, without worsening the quality of the results.

## 4.6 Luminance and chrominance interpolation for sub-pixel positions

As I have pointed out in previous sections, both techniques for disparity estimation require an interpolation tool. In the case of the pixel-based regularization approach, for example, a local average of neighbouring disparity vectors is used. This offers no guarantee that estimates will be integers and fall on points of the original sampling lattice. For example, in (4.18), the argument of $I_L(i + \alpha u, j)$ can arbitrarily fall anywhere in the left image. In the case of exhaustive search BM, sub-pixel precision of estimates (e.g., $q = \frac{1}{2}$) coupled with an arbitrary intermediate view position, $\alpha \in [0, 1]$, means that picture points not coinciding with the sampling lattice need to be determined. For example, in (4.4), the picture point given by $I_L^Y(m+\alpha\, d_{ij_x},\ n+\alpha\, d_{ij_y})$ does not have to belong to the sampling grid of the left image. Therefore, a means of interpolating image values at these arbitrary points from the existing picture points is required.

One well-known interpolator function is the *cubic convolution interpolator* discussed in [32]. The kernel associated with this interpolator is composed of piecewise-cubic polynomials defined on the subintervals (-2,-1), (-1,0), (0,1) and (1,2), and is zero outside this interval. In order to determine the coefficients defining the three cubic polynomials, certain conditions are applied. The kernel is required to be symmetric and continuous, and with a continuous first derivative. [10] Finally, the interpolation function which the kernel represents is to agree with the Taylor series expansion of the function being interpolated for as many terms as possible. From the above conditions emerges a unique kernel offering a third-order approximation.

The cubic interpolator function is used for both disparity estimation techniques

[10] A direct consequence of this is that the interpolation coefficients become simply the sampled data points since the only non-zero contribution to the convolution is at argument 0.

described in this chapter. For more details on its implementation, the reader is referred to [32].

# Chapter 5

# Intermediate view reconstruction

The previous chapter focused on developing two robust disparity estimation algorithms which provide very accurate solutions to the stereoscopic correspondence problem. As per the presented method for intermediate view reconstruction, the estimated fields are now used to reconstruct the final virtual image at position $\alpha$ (see Figure 4.2).

The chapter begins with a description of how image reconstruction is done, followed by view reconstruction results for both the block- and pixel-based techniques of Chapter 4. The quality of reconstructed views is quantified through the peak prediction gain (PPG) measure, which is defined in Section 5.3. Image reconstructions are included throughout the chapter, but since the printing process removes detail from images and hence poorly represents the quality of the reconstructions, all the results can be found in full colour at the following web page:
`<http://www.inrs-telecom.uquebec.ca/users/viscom/publications/>`.

Most image processing techniques require special attention at the boundaries of images, and the IVR problem is no exception. A method for eliminating visible distortions near image boundaries is proposed. Finally, a comparison between the block- and pixel-based approaches is offered, pointing out advantages, disadvantages and trade-offs of each technique.

## 5.1 Image reconstruction

To reconstruct a virtual, intermediate image, vectors from the estimated disparity field for a particular intermediate position are scaled as a function of $\alpha$ (see Section 4.2). Pixel $(i, j)$ in the virtual image $I_I$, may be computed using disparity-compensated linear filtering with a two-coefficient kernel. Here, we use a weighted average of the corresponding picture points in the left and right images, according to $\hat{d}_{ij}$, as follows:

$$I_I(i,j) = (1-\alpha)\, I_L\big(i + \alpha\hat{d}_{ij_x}, j + \alpha\hat{d}_{ij_y}\big) + \alpha\, I_R\big(i - (1-\alpha)\hat{d}_{ij_x}, j - (1-\alpha)\hat{d}_{ij_y}\big) \tag{5.1}$$

Non-linear filtering may also be used for image reconstruction. For example, picture point $I_I(i, j)$ may be reconstructed by taking either one of the corresponding picture points in the left or right image, based on some criteria. This is the "winner-take-all" approach, and it is motivated by the fact that since a *perfect* match is hard to find, the weighted average approach in (5.1) results in a blurred image reconstruction. The advantage of the "winner-take-all" approach is that without averaging, the detail of the original data is maintained in the intermediate image (no blur). The disadvantage is that neighbouring blocks, in a block-based approach say, could be reconstructed from different images. Hence, any luminance and/or chrominance differences in the stereoscopic pair could result in a "patchy" image reconstruction. In [28], a non-linear interpolation is implemented whereby only the left image is used for reconstruction if the position of the intermediate images is closer to that of the left image, and only the right image is used when it is closer to that of the right image. Occlusions are also handled in this approach.

The weighted-average approach results in a reconstructed image which is slightly blurred throughout, but in a block-based scheme, avoids patchiness resulting from image-pair mismatches. In addition, this approach preempts the need for developing a decision criterion to decide from which image, left or right, to reconstruct a given pixel in the intermediate image. For these reasons, the weighted-average approach is chosen here for the purpose of intermediate view reconstruction.

## 5.2 Image boundary handling

The model employed in intermediate view reconstruction proposed in Section 4.2 results in a negative side-effect for estimation of pixels falling on horizontal or vertical boundaries of the intermediate image. Consider the diagram in Figure 5.1 which shows a top view of the estimation of block $(i, j)$, located at the left edge of intermediate image "I".

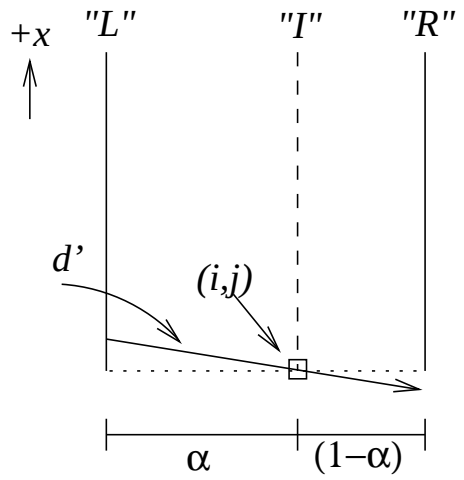



**Fig. 5.1** Top view showing the left, intermediate and right image planes. For vertical-boundary blocks, disparities with a non-zero $x$-component point outside either the left or right image planes, such as vector $d'$ which points outside the right image plane.

The side-effect is that all candidate disparity vectors for such a block should have a zero $x$-component since any other vector would result in one of its ends falling outside the boundaries of either the left or right image. This is a direct consequence of the fact that block $(i, j)$ must be a pivot point for all candidate disparity vectors. The same is true for blocks falling on the upper and lower boundaries of the intermediate image. Consequently, the correct disparity cannot be estimated for such blocks, resulting in visible image distortions in these areas.

To remedy this situation, estimation is carried out only within a sub-window of the full intermediate image. The first and last $M$ columns and rows of the disparity field are not estimated. After estimation, missing boundary vectors are replaced with the closest neighbouring vectors considered to be unconstrained. This is described

pictorially in Figure 5.2. For example, the top $M$ rows of vectors are replaced with the vectors from the $(M+1)^{th}$ row. This technique assumes that there are no significant depth discontinuities near image boundaries (smoothness constraint). Although this is perhaps not always a valid assumption – since depth discontinuities may occur anywhere – we have found that the technique works fairly well for the most part.

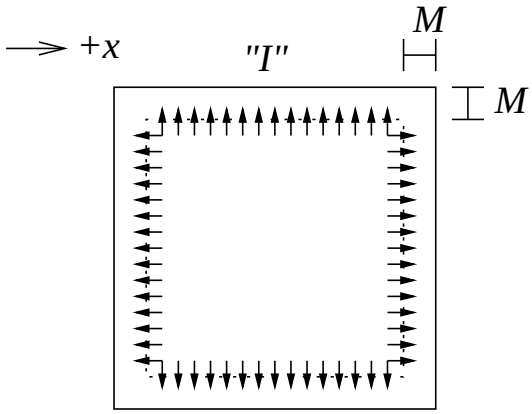
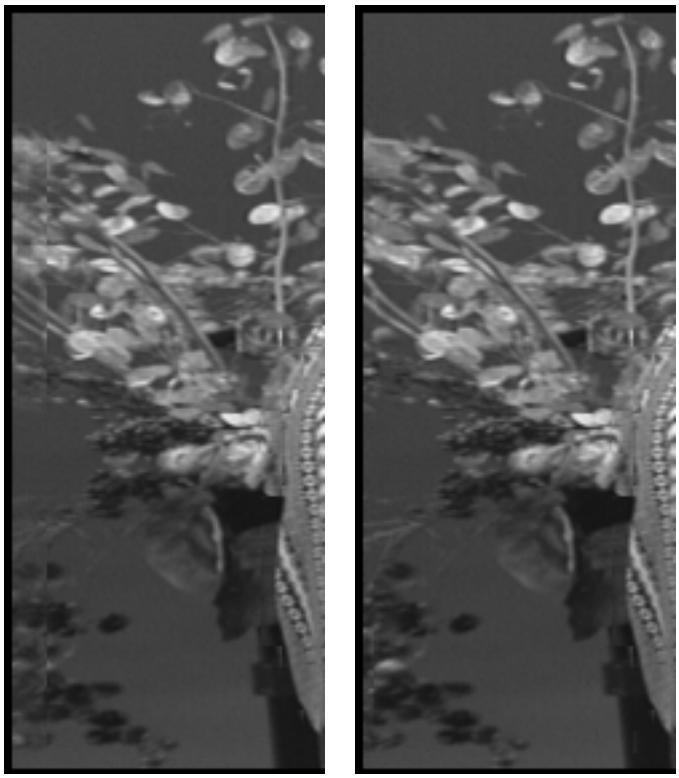



**Fig. 5.2** Vectors near image boundaries are replaced by neighbouring vectors to reduce image distortions in these areas.

The vectors which now replace those on the image boundaries are unconstrained, and some will inevitably fall outside the support of either the left or right image planes. Intermediate picture point $(i, j)$ is therefore reconstructed from only one of the corresponding picture points in a "winner takes all" fashion; the one which falls within the left or right image plane. For example, vector $d'$ in Figure 5.1 falls outside the right image plane, so pixel $I_I(i, j)$ is reconstructed using the corresponding point in the left image only, i.e., $I_I(i, j) = I_L(i + \alpha d_{ij_x}, j + \alpha d_{ij_y})$. This could not have been done earlier due to the nature of the estimation algorithms. Figure 5.3 compares the reconstruction of the left edge of the reconstructed intermediate image for test sequence *piano*; (a) shows the regular reconstruction suffering from boundary distortions, and (b) shows the result using the method described in this section. Notice the boundary distortions in the left column of pixels in (a) which are corrected in (b).

All reconstructions appearing in this chapter will use the image boundary-handling technique described here.



(a) without boundary-handling technique

(b) with boundary-handling technique

**Fig. 5.3** Intermediate view reconstructions for $\alpha = 0.5$ of test sequence *piano*, showing effect of boundary-handling technique on the left edge of the reconstructed image.

## 5.3 Quality assessment of image reconstructions

In this section, the impact of the enhancement schemes applied to the disparity estimation algorithms of Chapter 4 will be shown. Since original data is unavailable in the range of $\alpha \in (0,1)$, the quality of image reconstructions will be quantified through the peak-prediction gain (PPG). The PPG is a measure of the difference between original and predicted images. By reconstructing "virtual" views for $\alpha = 1.0$, we can consider the intermediate view as the *predicted* right image, and compare it to the *original* right image. This will give an idea on the quality of the reconstructed views. Note that the image boundary-handling technique of the previous section is a post-processing technique to reduce visible distortions, and should be excluded from the computation of the PPG. Therefore, the PPG is computed from the reconstructed images which have the the problematic boundaries.

The PPG is defined as follows:

$$PPG = 10 \log_{10} \frac{255^2}{\mathcal{E}} \quad [\mathrm{d}B], \tag{5.2}$$

with the mean-squared prediction error $\mathcal{E}$ given by

$$\mathcal{E} = \frac{1}{K} \sum_{(m,n)} \left[ I_r(m,n) - \hat{I}_r(m,n) \right]^2, \tag{5.3}$$

where $\hat{I}_r(x,y)$ is the reconstructed image at $\alpha = 1.0$, and $K$ is the number of pixels in the image. PPG results will be given for each image component, i.e., for $Y$, $U$, and $V$.

It is important to underline that the PPG measure is normally used to assess the performance of a prediction estimator. For example, it is often used to gauge the quality of match between two images related by disparity compensation. As has already been pointed out, the goal here is not prediction, hence we are not concerned with the absolute PPG numbers, but rather with the relative change of PPG due to each applied BM algorithm enhancement. Since for $0 < \alpha < 1$ there is no original data to compare the reconstructions with, the only way to assess the quality of the *intermediate* view reconstructions is by subjective evaluation. For simplicity, the following abbreviations are used to denote the various algorithm modifications

**BM1** ≡ 1-D (horizontal only) exhaustive search luminance-only BM,
**BM2** ≡ 2-D exhaustive search luminance-only BM,
**BM2B** ≡ 2-D exhaustive search luminance-only BM on luminance-balanced images,
**BM2B3** ≡ 2-D exhaustive search three-component Y-U-V BM on luminance-/chrominance-balanced images,
**BM2B3S** ≡ 2-D exhaustive search three-component BM on luminance-/chrominance-balanced images, using a smoothness constraint.

| | *flower* | | | *piano* | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **Y** | **U** | **V** | **Y** | **U** | **V** |
| **BM2** | +0.60 | +0.58 | +1.07 | +0.37 | −0.15 | −0.37 |
| **BM2B** | +0.51 | +0.46 | +1.34 | +1.62 | +0.98 | +1.32 |
| **BM2B3** | +0.45 | +1.00 | +1.83 | +1.57 | +1.21 | +1.66 |
| **BM2B3S** | +0.39 | +1.02 | +1.85 | +1.55 | +1.20 | +1.62 |

Table 5.1 Impact of proposed BM algorithm enhancements on the peak prediction gain (PPG) for image reconstructions at $\alpha = 1.0$. Results are relative to the base algorithm (exhaustive search 1-D BM), and are expressed in units of [dB].

Table 5.1 shows the change in PPG offered by each algorithm enhancement proposed in Chapter 4, and for each image component. The changes are relative to the base algorithm, simple 1-D exhaustive search block matching (BM).

From the numbers in Table 5.1, one notices in particular that using a 2-D exhaustive search offers significant gains in PPG for the luminance components of both test sequences. Changes in PPG for $U$ and $V$ in this case are really just side effects since the estimation is based on luminance only for **BM2**. The gain is greater for *flower* than it is for *piano* since it has a greater amount of vertical screen parallax. For **BM2B**, one notices a significant gain in PPG for the $Y$ component of *piano* (1.62dB), but a smaller change for *flower* as compared to **BM2**. This is because *piano* suffers from a much greater global luminance mismatch than *flower*. Again, for **BM2B**, estimation is still based on luminance only. In **BM2B3**, a three component match is performed, and the gains in PPG for $U$ and $V$ for both test sequences are high. Finally, for **BM2B3S**, although increases in PPG are high as compared to the

base algorithm, one obtains negligible changes in PPG as compared to **BM2B3**. As we will see in a later section, regularization via smoothness nevertheless has a great impact on the subjective quality of reconstructed views, especially in areas of low texture. Distortions in such areas will have a weak impact on the PPG, but a potentially strong negative impact on the quality of reconstruction.

As a general note, the quality of reconstructed views has been subjectively evaluated for each proposed algorithm enhancement, and their positive contributions to the quality of reconstructed views outweigh the increased complexity they induce. However, as the *law of diminishing returns* dictates, improvements are easiest to obtain in the beginning; as the quality of the reconstructed image improves, attainable gains diminish with each additional algorithm enhancement.

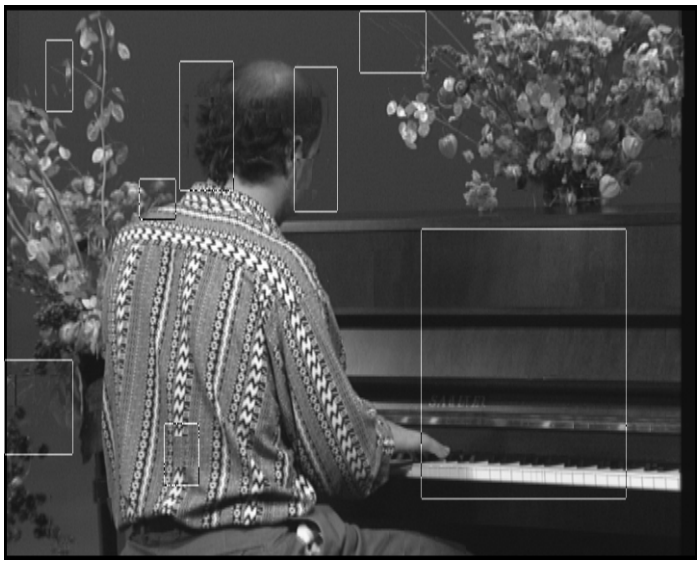
### 5.3.1 Block-based methods

In Chapter 4, the block matching algorithm was adapted to the problem of estimating a disparity field for an arbitrary intermediate view position. Several improvements were proposed to the basic BM algorithm, each with the aim of providing a more accurate vector field. This section looks at actual view reconstructions, pointing out the effects of the proposed enhancements on the reconstructed image quality.

The ordinary, exhaustive search block matching scheme was seen to produce disparity fields which were globally accurate. The image reconstructions for intermediate view position $\alpha = 0.5$ based on these vector fields (Figure 4.4) are shown for both test sequences in Figure 5.4 at full aspect ratio.

In general, the position of objects in the reconstructed views is correct, and the reconstruction offers the appropriate perspective view for $\alpha = 0.5$. For example, the position of the piano player's body in the intermediate view is exactly halfway between the corresponding positions in the left and right views. This is what one would expect for an intermediate position of $\alpha = 0.5$. However, although this initial reconstruction serves as an excellent starting point, there are obvious distortions in the reconstructed images. A few examples of problem-areas in the images are pointed out with highlighted rectangles in Figure 5.4. To name a few, notice the discontinuous pipes in the background wall of Figure 5.4(a) (refer to the original in Figure 4.1). Look at the distortions in the window located in the top left corner of this same image.

(a) *flower*



(b) *piano*

**Fig. 5.4** Intermediate view reconstructions, $\alpha = 0.5$, using the base algorithm, 1-D exhaustive-search block matching. State space defined by a 32-pixel search range. Example problem-areas are highlighted.

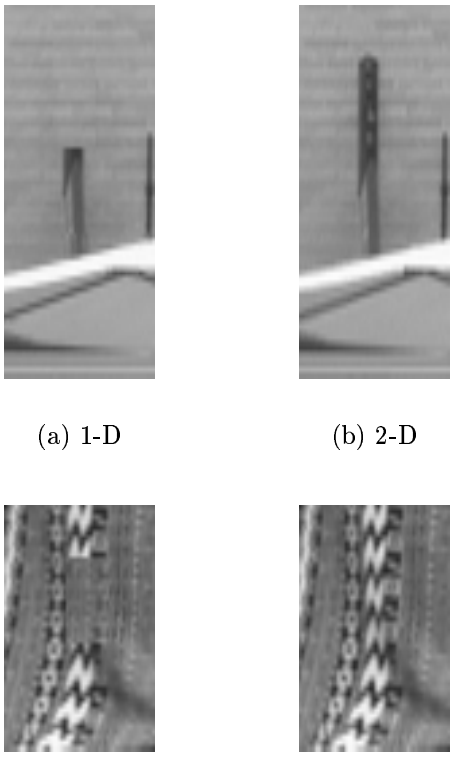
In (b), notice the obvious inconsistencies around the piano player's head and near the bottom of his shirt, in the middle. Also, the large area in the right half of the image made up mostly of the low-texture piano region contains blocking artifacts. The distortions are not very visible because of loss of detail from printing, but the full-colour images are available from the INRS VisCom web page mentioned previously.

The ambiguous matches which have been seen to exist in the estimated disparity fields in Chapter 4 are the causes of these distortions in the reconstructed images. As we will see, the proposed enhancements to the base block matching algorithm increase the overall accuracy of estimation, and reduce distortion in the reconstructed views. Rather than showing full image reconstructions for each proposed algorithm enhancement, I will show relevant portions of the reconstructed views which point out experimental improvements offered by the approach. Where appropriate, some of the image portions in the following sections will have modified grey-scale level distributions in order to increase the dynamic range of regions of interest and better highlight certain parts of the image.

The first algorithm enhancement was the inclusion of a vertical disparity component, which gave rise to a 2-D exhaustive search algorithm. This was seen to increase the complexity of the disparity estimation, but resulted in a more accurate vector field. Consider the missing pipe in the middle of the background wall, near the top boundary of the image reconstruction for *flower* in Figure 5.4(a). This portion of the image is enlarged in Figure 5.5. The pipe is discontinuous and not well reconstructed in (a), as the horizontal search resulted in an incorrect scalar disparity vector for the block of pixels forming the top part of the pipe. The problem is corrected with the inclusion of a 2-D search, as shown in (b). A similar situation exists for the image reconstruction of *piano*, shown in Figure 5.5(c) and (d). In (d), the portion of the piano player's shirt is well reconstructed.

There are a number of instances where a situation similar to that in Figure 5.5 occurs in image reconstructions for both test sequences, and as numbers in Table 5.1 confirm, overall, the inclusion of a vertical disparity component is beneficial to the reconstruction quality.

The second proposed algorithm enhancement is the estimation based on luminance-balanced images. To demonstrate the effect of this modification, consider the cut-outs



(a) 1-D (b) 2-D

(c) 1-D (d) 2-D

**Fig. 5.5** Enlarged portions of image reconstructions of *flower* and *piano* using algorithm **BM2** for $\alpha = 0.5$. Portions are cut out at position $(270, 20)$ for *flower*, containing the pipe in the background wall, and $(164, 428)$ for *piano*, containing the part of the piano player's shirt. The reconstructions in (b) and (d) are improved due to a 2-D exhaustive search.

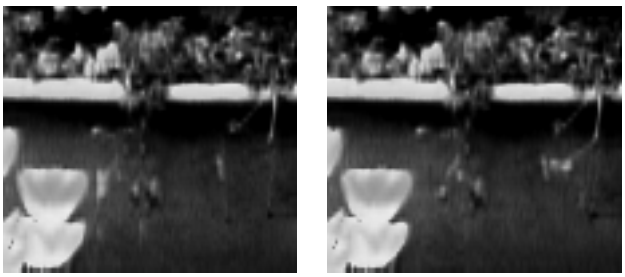
in Figure 5.6 for the reconstruction of test sequence *piano*. Notice the rather significant improvements around the outline of the head in (b). Although small problems persist in (b), these are easily corrected with inclusion of spatial smoothness constraint. Test sequences that stand to gain from this technique are those that have noticeable global luminance-mismatches, such as *piano*.

(a) estimation using **BM2** (b) estimation using **BM2B**

**Fig. 5.6** Enlarged portions of image reconstructions using algorithm (a) **BM2** and (b) **BM2B** of *piano* for $\alpha = 0.5$. Portions are cut out at position $(180, 20)$, containing the piano player's head. The reconstruction in (b) is drastically improved due to the estimation now based on balanced images.

The next algorithm enhancement is designed to tackle areas of the image where either luminance or colour information is low. By forcing a three-component match, a better overall image reconstruction is expected. Estimation is done on luminance- and chrominance-balanced images. Figure 5.7 shows portions of the reconstructed intermediate image for *flower* at $\alpha = 0.5$. In (a), the current luminance-based reconstruction is shown, and in (b), the full three-component match is shown. A better

match is obtained in (b). In some areas of the image however, by forcing a Y-U-V match, one actually obtains slightly worse results. Nevertheless, the enhancement has been tested and informally evaluated on several test sequences, and the conclusion is that overall, it is beneficial to the quality of the reconstructed images. The gains in PPG offered by this approach seem to agree with this.



(a) Y-only match (b) Y-U-V match

**Fig. 5.7** Enlarged portions of image reconstructions of *flower* using algorithm (a) **BM2B** and (b) **BM2B3** for $\alpha = 0.5$. Portions are cut out at position $(45, 240)$, consisting of the rim of the large foreground flowerpot. In (b), forcing a three-component match has eliminated small ambiguities in the obtained match.

The final algorithm enhancement which is proposed before any adjustment to the support of the model is made is the addition of a smoothness term in the minimization. Regularization generally has a great effect on the realism of reconstructed views since it is based on a very reasonable physical assumption, that the coherence of matter tends to give rise to smoothly varying characteristics in real-world scenes. The quality of image reconstructions for both test sequences using algorithm **BM2B3S** shown in Figure 5.8 is high. They may be compared to the original reconstructions obtained with simple BM at the beginning of this section. In particular, notice the improvements in the highlighted problem-areas of Figure 5.4. Few serious image distortions remain in Figure 5.8, and the intermediate view for position $\alpha = 0.5$ is quite acceptable.

Some of the problems that remain in the current image reconstructions are due to the too large support of the block-based model. Poorly reconstructed object bound-

(a) *flower*

(b) *piano*

**Fig. 5.8** Intermediate view reconstructions using the **BM2B3S** algorithm for $\alpha = 0.5$. The smoothness constant used is $\lambda = 15$.

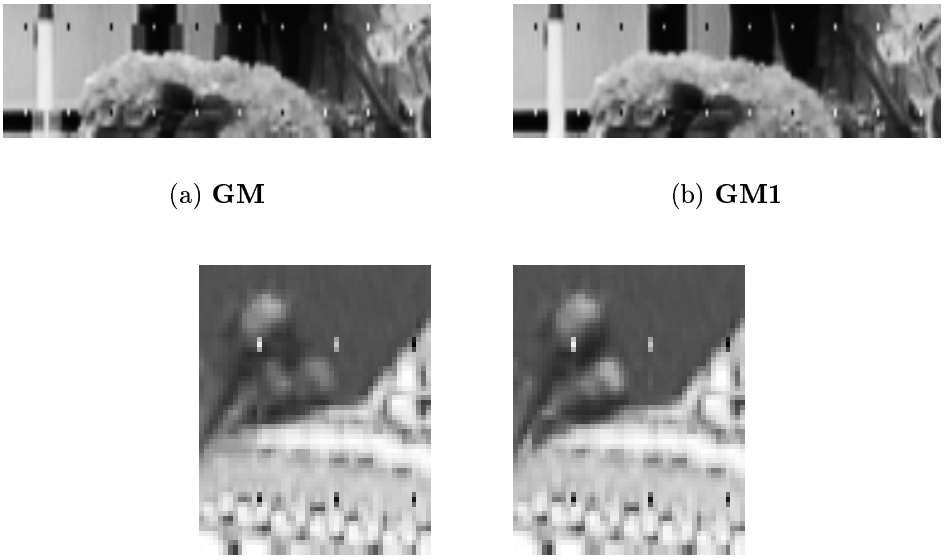
aries are obtained in areas where blocks of the intermediate image cover depth discontinuities. To better reconstruct object boundaries, the quadtree structure was adopted in Section 4.3.7. A procedure by which problematic blocks are detected, split, and then the sub-blocks reestimated was employed. An algorithm for the automatic detection of problematic blocks was developed by exploiting the characteristics of the robust estimator. Robust estimation was carried out by replacing the MAD criterion, characterized by a 0% breakdown point, with the Geman-McLure (GM) function, boasting a 50% breakdown point.

An approach for the recursive reestimation of the sub-blocks of targeted problematic blocks was presented. To show the effect of this algorithm, consider the example in Figure 5.9 which compares robust estimation with no splitting and robust estimation with one level of splitting. Two different scenarios are shown. To give an idea of the locations of blocks in each intermediate image, the pixels at each of the four corners of blocks are highlighted[1]. In the reconstruction for *flower* in (a), certain blocks are seen to fall on the overlap region of the foreground flowers and the black vertical bars in the background. The disparity vectors obtained for these blocks favour the flowers, since the reconstruction of the black bars is poor. The distortions formed by the vertical gray bars are due to the averaging of the black with the white counter, as a result of the incorrect disparity vector for the pixels in the upper regions of these blocks. In (b), one can see that quadtree splitting has allowed the different regions contained within the blocks to be well matched; objects are properly lined up, and one obtains a good reconstruction for both the foreground flowers, and the background dark vertical bars.

Similarly, in the reconstruction for *piano* in (c), one can see a faint doubling of the flower petal due to the fact that the block contains both the player's shoulder and the flower, the shirt being favoured for matching. In (d), splitting has allowed the flower petal to detach itself from the piano player's shoulder and obtain a good match.

To assess the performance of the quadtree structure for one and two levels of splitting, we first define the following abbreviations:

[1]Remember that images are interlaced and shown at full aspect ratio, even though only one set of lines – even or odd – is processed. This is why blocks are twice as high as they are wide.



(a) **GM** (b) **GM1**

(c) **GM** (d) **GM1**

**Fig. 5.9** Enlarged portions of image reconstructions of *flower* and *piano* for $\alpha = 0.5$ using algorithm (a)/(c) **GM** and (b)/(d)**GM1**. Portions are cut out at position $(360, 280)$ for *flower*, and (148,168) for *piano*. In (b) and (d), one level of quadtree splitting improves the reconstruction of object boundaries.

**GM** $\equiv$ robust estimation via Geman-McLure cost function, no splitting,
**GM1** $\equiv$ robust estimation via Geman-McLure, one level of splitting,
**GM2** $\equiv$ robust estimation via Geman-McLure, two levels of splitting.

Algorithms **GM, GM1** and **GM2** use the previously mentioned enhancements employed by algorithm **BM2B3S**. However, algorithm **GM** results in a slight decrease in PPG as compared to the **BM2B3S** algorithm. Although we would like the PPG measure to be proportional to how subjectively pleasing an image reconstruction might be, this is not always the case. While the image reconstruction obtained from the **GM** algorithm was judged to be of a higher quality than that obtained from **BM2B3S**, the decrease in PPG suggests otherwise. This is why subjective evaluation plays an important role in analyzing image data, since one cannot depend on a numerical measure to gauge the quality obtained.

For both image reconstructions for *flower* and *piano* at $\alpha = 1.0$, one obtains an increase in PPG for each image component when using the **GM1/GM2** algorithms vs. the **GM** algorithm with no splitting. The increases in PPG are given in Table 5.2. One would expect the increases to be relatively small, since only a fraction of blocks in the intermediate image are selected for reestimation. The main observation is that there is no decrease in PPG.

| | *flower* | | | *piano* | | |
|---|---|---|---|---|---|---|
| **Algorithm** | **Y** | **U** | **V** | **Y** | **U** | **V** |
| **GM1** | +0.19 | +0.15 | +0.05 | +0.26 | +0.04 | +0.09 |
| **GM2** | +0.21 | +0.18 | +0.07 | +0.28 | +0.04 | +0.09 |

Table 5.2 Increases in PPG (dB) for image reconstructions done at $\alpha = 1.0$, for one and two levels of quadtree splitting (algorithms **GM1/2**) vs. no splitting (algorithm **GM**).

To improve the reconstruction of boundaries even more, further levels of quadtree splitting may be performed. Figure 5.10 shows the evolution of a certain block of the image reconstruction for *flower* for two levels of quadtree splitting. Although the results are subtle, one can nevertheless make out the effect of one and two levels of splitting. In (a), no splitting results in a poor match for the foreground flowers (forming the lower quarter of the block defined by the four pixel-corners – the flowers

are blurred), and a good match for the white vertical bar which is the right edge of a window frame. Remember that the original block-size used is 16x16. In (b), splitting has resulted in the top two 8x8 sub-blocks, which fall entirely within the background wall, to be well-reconstructed. However, the lower two 8x8 sub-blocks still overlap the flowers, and require further splitting according to the quadtree procedure. Clearly the flowers dominated the search for these sub-blocks since the vertical bar is now quite distorted near this object boundary. The algorithm for automatic detection of problematic blocks is successful at selecting these lower sub-blocks for further splitting, and the reestimation of *their* 4x4 sub-blocks results in the reconstruction in (c). Although it is not obvious from the figure, (the reconstruction might look more accurate in (a)), the blurred flowers in (a) have become sharp and well reconstructed in (c). The algorithm has performed as desired.

(a) (b) (c)

**Fig. 5.10** Enlarged portions of image reconstructions of *flower* for $\alpha = 0.5$. Portions are cut out at position $(135, 120)$, and show the evolution of this image region for (a) zero, (b) one and (c) two levels of splitting.

We have experimented with splitting down to blocks of size 4x4 and even 2x2, and have noticed that the algorithm for automatic detection of problematic blocks remains robust down to blocks of size 8x8 only. With blocks of smaller sizes, the algorithm begins to be less reliable since there are very few samples on which to base the splitting decision. Furthermore, the block matching recursive reestimation algorithm, even with strong regularization, is not very robust for such small block sizes. We have found that two levels of splitting (i.e., down to reestimation of blocks of size 4x4) is the best compromise for intermediate view reconstruction.

The algorithm for automatic detection of problematic blocks required a number

of threshold values (Table 4.3). Although the use of such thresholds is necessary, the approach was designed to be as insensitive as possible to the actual values assigned. The default values used, shown in the same table, were largely varied in an attempt to understand the (in)sensitivity of the algorithm to threshold value fluctuations. Experiments proved that the selection of threshold values is not crucial to the performance of the algorithm (values listed in Table 4.3 were used for all experiments). In essence, these threshold values affect only the number of blocks chosen for reestimation. Tagging an excessive number of blocks as problematic is not a serious problem, since the reestimation algorithm is robust enough not to worsen estimates for blocks that really did not require splitting. In addition, threshold values have to be set to highly inappropriate values for the algorithm to begin to perform poorly.

The final reconstructed images for $\alpha = 0.5$ for the block based approach, which encompasses all proposed algorithm enhancements, and which employs two levels of quadtree splitting, are considered to be subjectively very pleasing and accurate. I will not include the reconstructions here since, due to loss of detail from printing, they would not look any different from previous reconstructions obtained using the **BM2B3S** algorithm. However, the full-colour versions are available from the INRS VisCom web site cited previously.

### 5.3.2 Pixel-based methods

Previous sections examined the results obtained from the block-based approach to intermediate view reconstruction. In this section, image reconstructions obtained using the regularization approach described in Section 4.4 are examined. The disparity fields from this approach are pixel-based, meaning one vector-per-pixel is described. The approach was originally based on the paper by March in [18], and adapted to IVR. The only algorithm enhancement was the inclusion of a vertical disparity component in the minimization. The resulting disparity fields from Chapter 4 are highly regular, and accurate. Image reconstructions for both test sequences are shown in Figure 5.11.

Certain regions of the reconstructions in Figure 5.11 are highlighted with rectangles. In (a), the highlighted regions of the reconstruction for *flower* show how the pixel-based approach performs with respect to the same boundaries that the block-

(a) *flower*

(b) *piano*

**Fig. 5.11** Intermediate view reconstructions, $\alpha = 0.5$, using the pixel-based regularization approach. Smoothness constraint is applied with $\lambda = 500$, and a vertical disparity constraint with $\gamma = 1000$.

based approach had difficulty reconstructing (see Figure 5.4). Overall, objects are well reconstructed, even though smoothing across their boundaries was performed and is inherent to the algorithm. In (b), also notice how the large region made up mostly of the piano on the right part of the image is highly regular. Through comparison of the full-colour reconstruction with the original *piano* stereo pair, one can see that object-texture has been well maintained.

To evaluate the performance of the inclusion of a vertical disparity component, image reconstructions are done for $\alpha = 1.0$, and are compared with the original right images via the PPG measure. Changes in PPG are shown in Table 5.3. Notice how the enhancement has a greater effect on the image reconstruction for *flower* since this stereo pair has greater vertical parallax values than *piano*.

| **Algorithm description** | *flower* | | | *piano* | | |
|---|---|---|---|---|---|---|
| | **Y** | **U** | **V** | **Y** | **U** | **V** |
| regularization with 2-D disparity vector | +0.54 | +0.29 | +0.38 | +0.26 | −0.20 | −0.12 |

Table 5.3 Changes in PPG (dB) for image reconstructions done at $\alpha = 1.0$, for scalar vs. two-dimensional disparity vectors using the pixel-based regularization approach.

As an overall comparison of the block- vs. pixel-based approaches, consider the PPG gains in the luminance components of the reconstructions obtained using the pixel-based approach over those obtained from the **GM2** algorithm shown in Table 5.4.

| *flower* | *piano* |
|---|---|
| **Y (dB)** | **Y (dB)** |
| +1.05 | +0.43 |

Table 5.4 Gains in PPG (dB) for the luminance components of image reconstructions done at $\alpha = 1.0$ using the pixel-based approach (2-D vectors), as compared to the **GM2** block-based approach.

### 5.3.3 Discussion

Two algorithms for disparity estimation have been examined. Both have proven to give accurate solutions to the stereoscopic correspondence problem in the context of intermediate view reconstruction. This section offers a discussion on the performance of these algorithms, pointing out their relative complexity, quality of image reconstructions and their feasibility in terms of practical video applications.

The block-based approach is conceptually simple, which is what makes it so attractive. The 1-D exhaustive search block matching scheme (the so-called base algorithm) has already been implemented in hardware for real-time applications today [33]. However, this simple template-matching scheme resulted in a set of initial reconstructions which had numerous distortions, as the first image reconstructions shown in this chapter confirm. In order to improve the quality of reconstructions, the problems were identified and corrected through various algorithm enhancements. By the end, with the adoption of the iterative quadtree structure to improve boundary reconstructions, the once simple block matching scheme had become very complex. The algorithm is computationally intensive, and demands complex data structures and housekeeping to implement in software.

In contrast, the pixel-based regularization approach is conceptually more complex. It is based purely on mathematical relationships, and no special attention was given to improve the quality of object boundary reconstructions. The simple image boundary handling technique of Section 5.2 was, however, applied. At first, this approach was computationally more taxing that the initial simple block matching scheme. However, with all the improvements made to the BM algorithm and the increased complexity that went with them, the software program implementing the regularization approach now takes less time to execute.

In terms of the quality of image reconstructions, the block-based approach, with all algorithm enhancements, offers excellent results. The image reconstructions maintain object structure in the original stereo pair very well. This is a direct consequence of the fact that the algorithm is block-based. Furthermore, the results offer excellent representations of different perspective views. The main problems with the approach, however, are in dealing with disparity discontinuities within blocks.

The image reconstructions from the regularization approach applied to various

test sequences are also of a high quality. This pixel-based algorithm, however, suffers from other artifacts known as *object-warping* which are due to over-smoothing. Since the pixel-based approach does not match templates or blocks, only single pixels, it does not maintain object structure quite as efficiently as the block-based scheme. In summary, the overall subjective quality of image reconstructions is comparable for both approaches.

It is interesting that the only modification made to the regularization approach was the inclusion of a vertical disparity component. Otherwise, it is really just the approach of March [18] adapted to IVR. Nevertheless, it offers a significantly higher PPG for the luminance components of both test sequences as compared to the best obtainable BM results. This is because the pixel-based approach is more flexible in that, unlike the block-based approach, it allows small variations to exist between neighbouring vectors. The lack of 100% smoothing within a block of pixels results in a more accurate reconstruction, hence a higher PPG. Furthermore, since this pixel-based approach is less intensive computationally, it remains for now the obvious choice for any real application in intermediate view reconstruction.

## 5.4 Practical applications for IVR

Throughout this thesis, we have spoken exclusively about reconstructing a single field from a stereo pair at a particular lateral position, $\alpha$. In a practical application, however, a continuum of reconstructed views is needed, e.g., as a function of the viewer's head position with respect to the screen. Clearly, a continuous sequence of images must be reconstructed. A simple approach is to reconstruct every image of this sequence independently of others. This is what I have implemented here, but as discussed below, certain dynamic distortions become visible.

In the context of a block-based scheme, one can imagine a scenario where due to the geometry of the scene, a particular block is problematic in the image reconstruction at position $\alpha_1$, but the same block is well reconstructed in the virtual image at position $\alpha_2$. Consequently, the viewer may see distortions from one viewing angle, but not the other, and the perspectives are inconsistent. In the context of a pixel-based scheme, a similar situation may exist where disparity estimation at two different $\alpha$-

positions results in conflicting estimates and hence different-looking representations of the same scene. In order to avoid such a situation, a more advanced and complex approach would be to perform the reconstructions with a disparity constraint between neighbouring reconstructed fields. The next section explores various ways of reconstructing a continuum of $\alpha$'s.

### 5.4.1 Reconstruction for a continuum of $\alpha$'s.

The intermediate view reconstruction must be carried out on a 3-D sampling lattice[2]. The first two coordinates are given by the spatial position of the sampling point within the intermediate image, depicted in Figure 5.12. The third coordinate is given by the position of the intermediate view, i.e., $\alpha$.

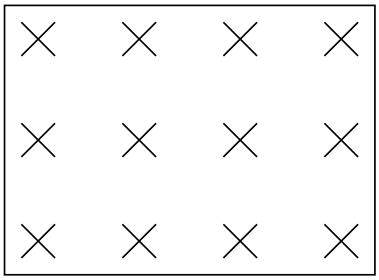


**Fig. 5.12** Crosses show points of the sampling lattice defined on the plane of the intermediate image. This is an example of a digital image with four pixels horizontally, and three vertically.

#### Method 1 – reestimation

In principle, a separate disparity estimation procedure must be carried out for each desired intermediate view position (each $\alpha$). In the event that a set of disparity fields for a range of intermediate view positions is desired, this method is a very costly procedure, although theoretically optimal. Consequently, a more efficient solution is sought.

[2]A lattice is a regular array of points in space and time, defining the discrete positions at which the colour and intensity of a digital image are specified.

Furthermore, since in such applications as 3-D video-conferencing or 3-D TV, images are coded to exploit cross-view redundancy, disparity information for $\alpha = 0$ or $\alpha = 1$ is typically transmitted along with the coded reference image. Thus, it would be beneficial to perform an $\alpha$-continuous IVR based on a single disparity field. Below, we define two other ways of obtaining disparity fields for any arbitrary position given only one field, estimated at some other position.

### Method 2 – non-compensated propagation of disparities

The method based on non-compensated propagation of disparity vectors consists of estimating a disparity field for some intermediate view position $\alpha_1$, and propagating this vector field onto the sampling lattice of $I_2$, defined at position $\alpha_2$, without disparity compensation. This is shown pictorially in Figure 5.13.

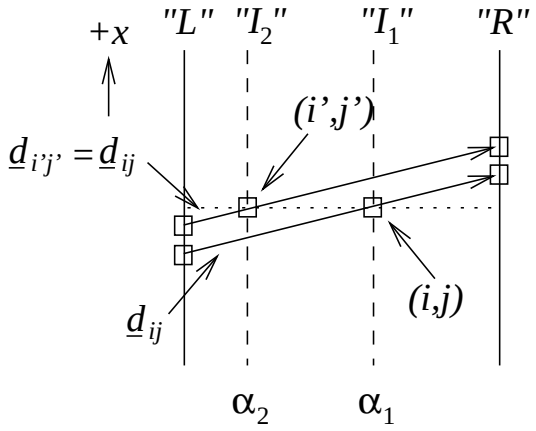



**Fig. 5.13** The method of non-compensated propagation of vectors for obtaining a disparity field for a particular intermediate view position, $\alpha_2$, from the estimated field of some other position, $\alpha_1$. The disparity vector for token $(i, j)$ in $I_1$ is propagated to the token in $I_2$ at the corresponding position, $(i', j') = (i, j)$, giving $\underline{d}_{i'j'} = \underline{d}_{ij}$.

The disparity vectors specified in the estimated vector field for $\alpha_1$ are used to reconstruct the desired intermediate view position at $\alpha_2$. Consistent with our model,

$I_2$ is obtained by using the following well-known relationship:

$$I_I(i,j) = (1-\alpha_2)\, I_L(i+\alpha_2 d_{ij_x}, j+\alpha_2 d_{ij_y}) + \alpha_2\, I_R(i-(1-\alpha_2)d_{ij_x}, j-(1-\alpha_2)d_{ij_y}), \tag{5.4}$$

where $d(\cdot,\cdot)$ is defined on the sampling lattice of the intermediate image plane at $\alpha_2$.

This method is based on the approximation that within a small range, neighbouring pixels at the same image-position but at different intermediate view positions should have similar disparity vectors.

### Method 3 – disparity-compensated propagation of disparities

Given a disparity field for intermediate view position $\alpha_1$, and defined on the intermediate image $I_1$, the disparity vectors for each token $(i,j) \in I_1$ are extended, and their intersections with the desired viewpoint's image plane, $I_2$, computed. The intersecting token at $(i',j')$ in $I_2$ inherits the disparity vector given by $\underline{d}_{ij}$, as shown in Figure 5.14.

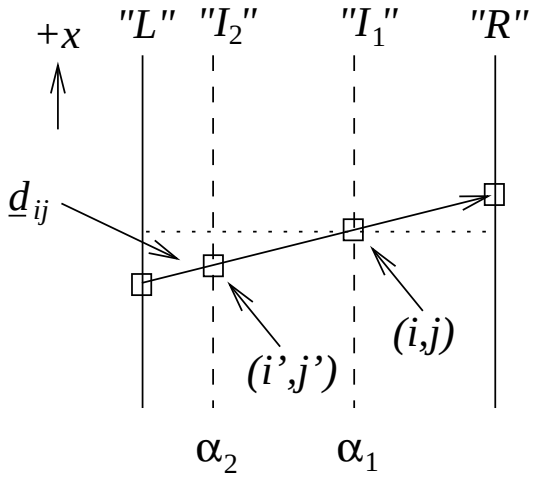



**Fig. 5.14** The method based on disparity-compensated propagation of vectors for obtaining a disparity field for a particular intermediate view position, $\alpha_2$, from the estimated field of some other position, $\alpha_1$. Disparity vectors defined in $I_1$ are extended, and their intersections with $I_2$, $(i'j')$, computed. The set of intersections should completely define $I_2$.

Given the nature of the model, this particular method offers no guarantee that

resulting intersecting tokens in $I_2$ will coincide with points of the desired sampling lattice. On the contrary, an arbitrary, highly non-uniform sampling grid is obtained, much like that in the example of Figure 5.15.

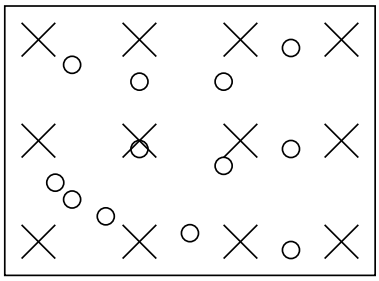


**Fig. 5.15** Crosses show points of the original sampling lattice defined on the plane of the intermediate image. Circles are intersecting points (disparity-compensated) forming a non-uniform sampling structure.

That said, a means of performing uniform-grid interpolation (the crosses) from non-uniformly spaced data (the circles) is required. One option could be that each sampling point of the original lattice inherits the disparity vector belonging to the closest intersecting point of the non-uniform sampling structure. This would require a robust set of rules for determining the closest intersecting point. Although it might seem simple in the example of Figure 5.15 as to which crosses to match with which circles, a more complex scenario is easily envisioned where the choice is not so obvious. This is no trivial task to implement algorithmically.

Alternatively, one could compute an average of the vectors belonging to two or three of the closest intersecting points. Either way, the operation is much too complicated, and is still inexact.

### Reconstruction of a continuum of $\alpha$'s – Discussion

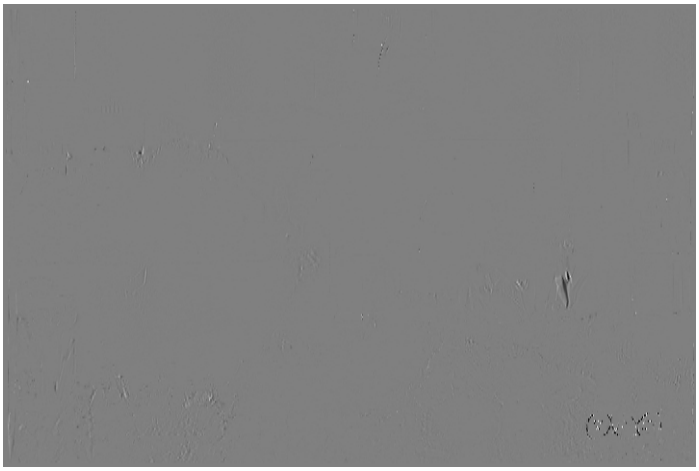
Of the three methods described herein, one would expect reestimation at each desired intermediate view position (Method 1) to give the most accurate results. Therefore, we will compare the view reconstruction results obtained using the method of non-compensated propagation, to those obtained from reestimation, which is very costly. A disparity field at $\alpha_1 = 0.5$ is computed for both test images using the pixel-based

regularization technique. Using the method of non-compensated propagation (method 2), this same disparity field is then used to reconstruct a view at $\alpha_2 = 0.25$, which we will call $I_{prop}$. In comparison, a separate disparity field is estimated for $\alpha_2 = 0.25$, and the corresponding view is reconstructed, $I_{reest}$ (subscript "reest" is an abbreviation for "reestimated", since this image is based on a disparity field which was reestimated for this particular intermediate view position). A *difference image* is computed for each test sequence, which is simply a component-wise difference between two images, here $I_{reest}$ and $I_{prop}$. The luminance component of a difference image is typically shifted by 128 for visibility. The result for both test images is shown in Figure 5.16.

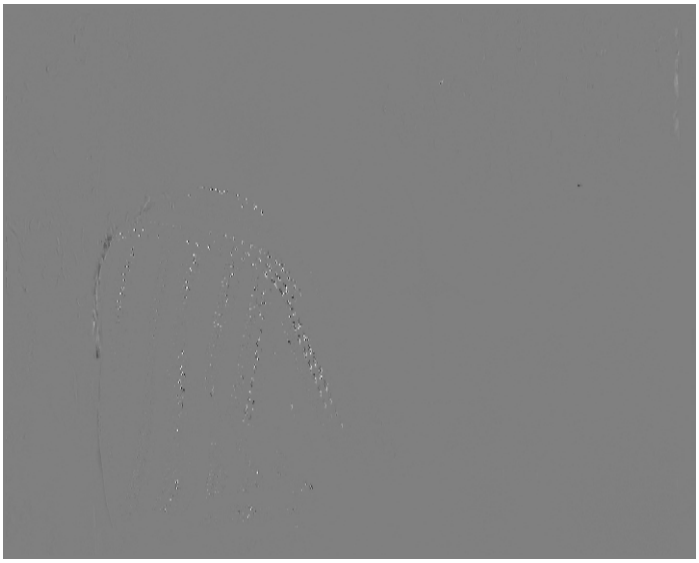
Since the difference images have high grey content (luminance of 128), i.e., very small errors, the method of non-compensated propagation of vectors for obtaining image reconstructions is seen to perform very well. Due to its extreme simplicity, it is positively surprising to see how small distortions it causes. Using this method, one estimated disparity field (typically at $\alpha = 0.5$) is enough to perform image reconstructions at any other arbitrary intermediate view position; and this, with small error as compared to reconstructions based on the recomputation of disparity for each desired viewpoint.

By estimating a single disparity field at $\alpha = 0.5$, the method of propagation has been used to reconstruct images in the range of $\alpha \in [-0.25, 1.25]$. Beyond this range, the reconstructions suffer from serious distortions. The performance of this method is also highly dependent on the magnitude of the disparities. The propagation of large vectors runs a higher risk of encountering occlusions. Furthermore, test sequences used in this work were acquired from closely positioned cameras, and capture faraway views. For close-ups, this method may cause problems.

As mentioned, in a conventional stereoscopic video transmission system, disparity information for $\alpha = 0$ or $\alpha = 1.0$ is available at the receiver. Therefore, it would be beneficial if one could reconstruct intermediate views based on this disparity information already available. To judge the feasibility of whether the method based on non-compensated propagation of disparity vectors (method 2) can be used in this case, we estimate disparity for intermediate view position $\alpha = 1.0$, and use the resulting vector field to reconstruct views at $\alpha = 0.25$, $\alpha = 0.5$ and $\alpha = 0.75$. We then compare the reconstructions to those obtained by estimating disparity directly at the



(a) *flower*



(b) *piano*

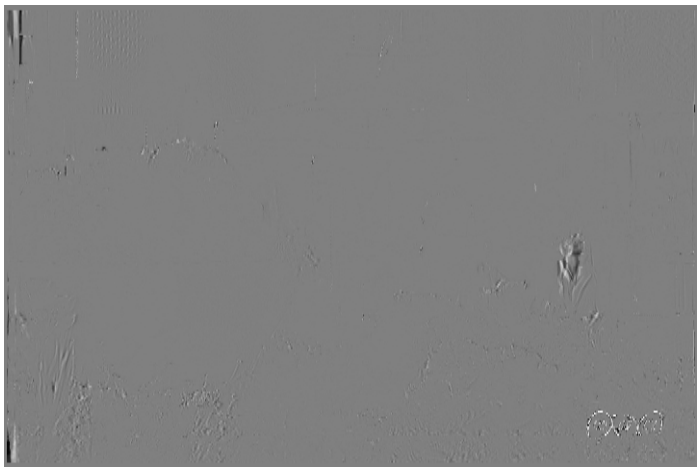
**Fig. 5.16** Difference images between two different intermediate view reconstructions at $\alpha = 0.25$. One reconstruction is done by propagating the vectors from the disparity field estimated for $\alpha = 0.5$ to $\alpha = 0.25$ (method 2), and the other is based on the actual estimated disparity field for $\alpha = 0.25$.

three positions. The difference images are shown in Figures 5.17 and 5.18.

In Figure 5.18, one can clearly see that the error in the difference images decreases as the position of the reconstructed view approaches that from which the estimation is based (here, $\alpha = 1.0$). Although this is also true for *flower* in Figure 5.17, the effect is less obvious. The reconstructions at $\alpha = 0.25$, which are based on the disparity field estimated for position $\alpha = 1.0$, are relatively less accurate than the reconstructions at $\alpha = 0.75$, obtained using the same method of non-compensated propagation of vectors. Intuitively, this makes sense since the further one tries to reconstruct from the starting point, the more occlusions pose a problem. The difference images in (c) for both test sequences have the lowest overall error.

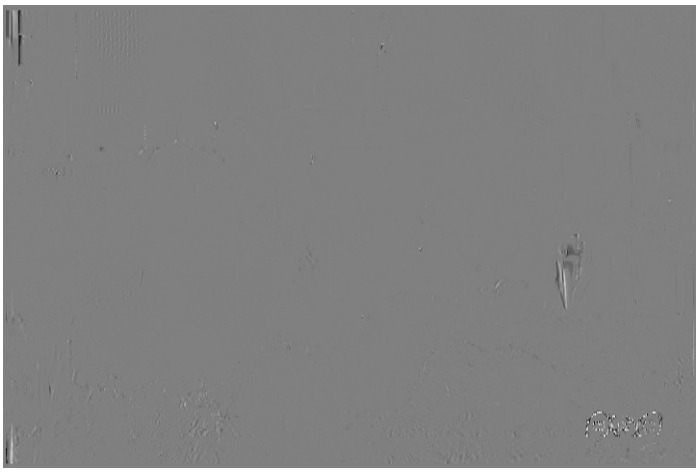
Nevertheless, in all cases there is high grey-content in the images (low error). The method therefore works reasonably well in reconstructing intermediate views based on disparity information available for the extreme positions of $\alpha = 0$ or $\alpha = 1.0$.

(a) *flower*, $\alpha = 0.25$



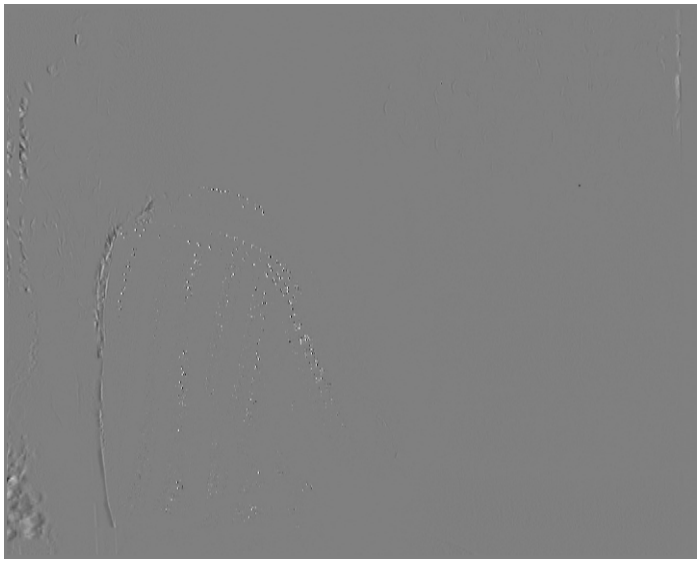
(b) *flower*, $\alpha = 0.50$

**Fig. 5.17**

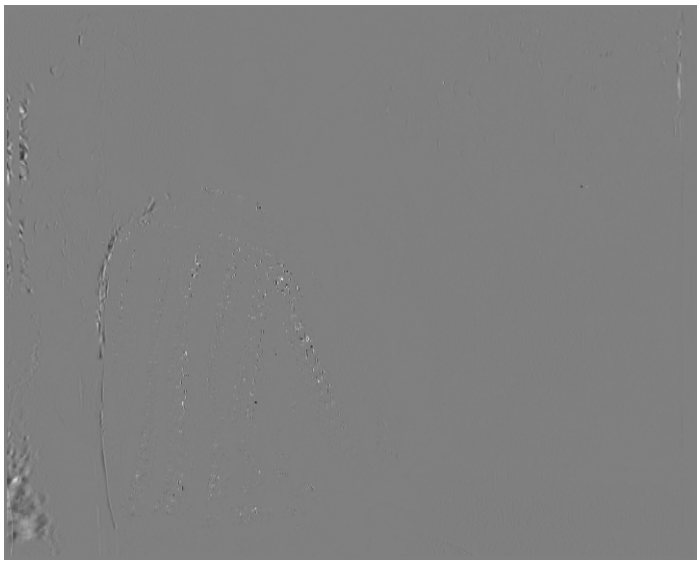


(c) *flower*, $\alpha = 0.75$

**Fig. 5.17** Difference images between two different intermediate view reconstructions for positions (a) $\alpha = 0.25$, (b) $\alpha = 0.5$ and (c) $\alpha = 0.75$. One reconstruction is done by propagating the vectors from the disparity field estimated for $\alpha = 1.0$ (using method 2), and the other is based on the actual estimated disparity fields for $\alpha =$ (a) 0.25, (b) 0.5 and (c) 0.75.

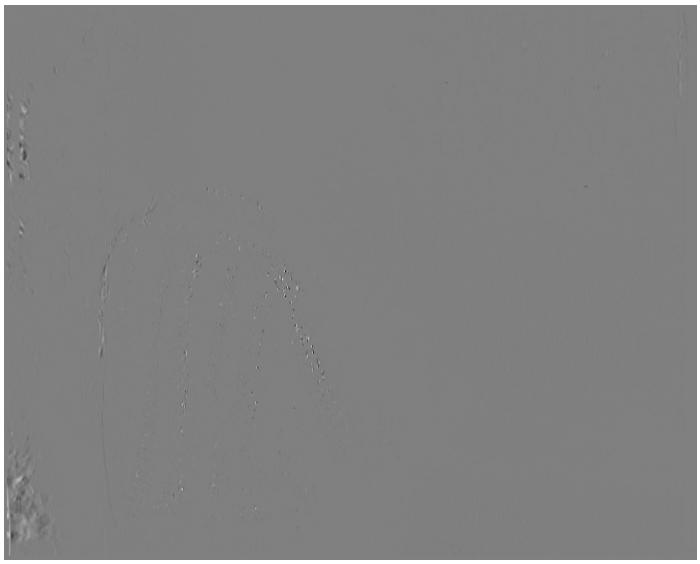


(a) *piano*, $\alpha = 0.25$



(b) *piano*, $\alpha = 0.50$

**Fig. 5.18**



(c) *piano*, $\alpha = 0.75$

**Fig. 5.18** Difference images between two different intermediate view reconstructions for positions (a) $\alpha = 0.25$, (b) $\alpha = 0.5$ and (c) $\alpha = 0.75$. One reconstruction is done by propagating the vectors from the disparity field estimated for $\alpha = 1.0$ (using method 2), and the other is based on the actual estimated disparity fields for $\alpha =$ (a) 0.25, (b) 0.5 and (c) 0.75.

# Chapter 6

# Summary and conclusions

Most of the work presented in this thesis focuses on 3-D video for entertainment purposes. Today, IMAX® Corporation of Mississauga, Ontario, for example, is enjoying international recognition for their high-quality, large-screen stereoscopic films. As the technology for stereoscopic video becomes even more popular and widely accepted, a demand for porting it to private television screens will develop. The reality of 3-D broadcast TV in the future is largely contingent on the development of solutions to existing problems with stereoscopic video today. The focus of this thesis is hence on the development of high-quality stereo image reconstructions, with intended application in the entertainment industry.

Some problems identified in this work are *parallax adjustment* and *continuous look-around*. The need for adjusting the screen parallax of a stereoscopic display stems from the fact that stereoscopic acuity among individuals varies a great deal. Indeed, I myself often have trouble fusing the stereo image of certain IMAX scenes and experience considerable discomfort. Similarly, continuous look-around is required in order to give viewers at any lateral position a realistic representation of the 3-D scene. In both cases, an intermediate view is required offering a different perspective of the scene. The desired perspective is a synthetic representation of an image captured by a virtual camera whose position is different than that of the original left and right cameras. In other words, the problems of parallax adjustment and continuous look-around are mostly made possible by intermediate view reconstruction.

## 6.1 Thesis summary

This thesis began with a background introduction to the field of stereoscopic imagery. A basic understanding of how the human visual system provides depth information to the brain through *retinal disparity* was developed. A single point in a three-dimensional scene projects itself onto two different locations on the retinae of the eyes. The difference in relative position between these two points is defined as the retinal disparity. The geometry of crossed and uncrossed disparity was presented, and then translated to the viewing screen. Screen parallax, or the difference in position of homologous points on the display, was seen to induce retinal disparity, which in turn provides depth sensation when viewed on a stereoscopic display.

Image acquisition, in the context of stereoscopic viewing for entertainment purposes, is carried out by two cameras placed side by side. The left and right cameras are aligned vertically, and separated by a horizontal distance about equal to the average adult interpupillary distance (64mm). This is called the *inter-axial separation*. The setup of the two cameras was discussed in the context of the *parallel* and *toed-in* configurations. The geometry of the parallel configuration is governed by simpler mathematical expressions, and does not induce any vertical disparity between the two acquired images. The inverse proportionality of object-depth vs. screen parallax was derived for the parallel cameras. In this configuration, typically the CCD sensors of the two cameras are horizontally shifted in order to increase the common field of view. On the other hand, the toed-in configuration is simpler to set up and often used in practice since it maximizes the common field of view between the cameras.

The acquired left and right images are then displayed on a viewing screen. Various methods exist for separating the left and right images and showing only the correct perspective to each eye. One popular technology used by both IMAX and INRS is the spatial superposition and temporal interleaving of the left/right images. Active liquid-crystal (LC) shutter glasses are then used which operate in sync with the display, and only allow the correct perspective view to be seen by each eye.

The distortions suffered by the various display technologies were presented. One of the most serious of these is *shear distortion*. Shear distortion is caused by the fact that typical stereoscopic displays which do not track viewer head movements can offer only one perspective view of a scene at a time. Lateral viewer head movements

result in a distorted representation of the scene when viewed from an angle which is different from the intended viewing angle. The ability to reconstruct intermediate views alleviates this problem by displaying the correct perspective view as a function of the viewing angle. In so doing, the viewer enjoys *continuous look-around* of the stereoscopic display.

Intermediate view reconstruction (IVR) also permits the adjustment of *screen parallax*. The inter-axial separation of the cameras has a large impact on the strength of the stereoscopic cue experienced by the viewer. A *fixed* camera separation results in a stereoscopic cue which is usually not suited to all viewers. Hence, by reconstructing intermediate views, the camera separation can be adjusted (increased or reduced), which in turn affects the amount of induced screen parallax on the display. Such a scenario will allow viewers to adjust the "3-D level" of the stereo image, much like typical computer monitors today allow viewers to adjust other parameters like contrast and brightness.

IVR also plays an interesting role when porting large-screen stereo images to a small screen. Here, the large disparities between perspective views that a large-screen display can afford (acquired from cameras with a large inter-axial separation), are no longer tolerable for small screens with smaller viewing distances. This is another example of the need for parallax adjustment. As well, IVR can be used for *missing-frame(view) replacement* in a multiview system.

The approach to IVR taken in this thesis is based on signal processing techniques. Unlike 3-D model-based techniques which typically perform arbitrary view generation for objects only, the approach here makes no assumption on image-content. In order to reconstruct an intermediate view, a mapping between the left and right images is first obtained. The mapping comes in the form of a vector field which describes the displacement of each pixel in the right image with respect to its corresponding position in the left image. The process of *disparity estimation* is used to solve this correspondence problem and obtain a disparity field. Then, disparity-compensated interpolation is used to reconstruct the intermediate views. While the interpolation is carried out in the usual manner, i.e., using a linear filter with angle-dependent coefficients, the disparity estimation algorithms are novel in several ways.

Two classes of disparity estimation techniques were explored; block-based and

pixel-based approaches. First, the typical exhaustive-search block matching (BM) scheme was adapted to estimate a disparity field for a specific intermediate view position. To do this, a model was adopted such that the resulting estimated vector field is defined on the plane of the intermediate image. Block matching schemes are known to be associated with a number of problems. In the context of IVR, these are identified and cured using various techniques. Among them are colour-based estimation, spatial smoothness constraint and adoption of a robust cost function. A technique for eliminating global luminance and chrominance mismatches between the left and right images is also presented.

Since block-based schemes assign the same disparity vector to all pixels in a block, problems arise, for example, in blocks that cover regions of object-overlap. In such cases, if the overlapping objects belong to different depth planes, one vector is insufficient to describe the correspondence of all member-pixels. To tackle this problem, a technique for the automatic detection of such problematic blocks, which is based on robust estimation, is proposed. A quadtree structure approach is taken whereby the block is split into four equal-size blocks, and the sub-blocks are reestimated.

All techniques are shown to offer interesting gains in the accuracy of the estimated disparity fields. The BM scheme which encompasses all algorithm enhancements is tested on several stereoscopic test sequences, and results for two of these are presented. The reconstructed images properly portray the scene from a different viewing angle, and the overall quality of the still images is acceptable.

The pixel-based technique is based on the prediction-based disparity estimation algorithm introduced in [18], and adapted to perform IVR. The regularization term is modified to implement second-order smoothing, and since no assumption is made on the geometry of the cameras (parallel or otherwise), a two-component $(u, v)$ disparity vector is computed. Pixel-based schemes suffer from certain problems as well. For example, the regularization (smoothness) term typically results in a poor reconstruction of object boundaries due to over-smoothing. This is because the method usually enforces a strong likeness between neighbouring vectors. The block-based schemes tend to be superior in maintaining object structure. Nevertheless, the quality of obtained image reconstructions using the regularization approach is considered to be high, especially if a sequence of images is considered (no temporal discontinuities in

the reconstructions).

The image reconstructions resulting from the two approaches discussed in this thesis have been used to perform parallax adjustment of still stereo images. The virtual camera separation was adjusted from a normalized distance of 0 to 1, and beyond, in increments of 0.1. Indeed, the stereoscopic cue in the image was seen to vary as a function of the virtual camera separation, and no visible distortions were reported.

Furthermore, the range over which the method for computing intermediate views could be applied was explored. By estimating a disparity field for an intermediate view position located exactly mid-distance between the original left and right cameras, intermediate views were reconstructed for a wide range of positions. The approach was seen to offer acceptable to excellent results up to a range of 25% beyond the position of the original cameras, in either direction. Beyond this, occlusions result in unacceptable artifacts.

## 6.2 Future work

As discussed in the previous section, results from the pixel-based approach suggest that it is more suited to real applications in stereoscopic video where sequences of images are typically used. Although the block-based results are good for still images, playback of a sequence of reconstructed views reveals highly visible artifacts. The root of the problem is due to a lack of disparity vectors to properly represent all pixels in the image. The pixel-based approach, however, which represents each pixel by a separate vector, does not suffer from such distortions when a sequence of reconstructed views is created.

That said, results of this thesis suggest that the focus of future work in the field of IVR using 2-D signal processing techniques should be on further improving the pixel-based approach presented herein. The main problem with the current approach is that it does not perform selective smoothing. It would be beneficial to enforce smoothing only *within* objects of a 3-D scene. Otherwise, vectors belonging to pixels which are near object boundaries are forced to have similar vectors as those belonging to pixels from another object. If these two objects are located at different depths,

the result is a poor reconstruction of the boundary between them. In [18], March discusses the idea of "selective smoothing" by enabling regularization only between pixels belonging to the same object. This is a good idea, but the problem remains of establishing an image analysis tool which determines where objects are located in an arbitrary scene. This is no trivial task, and methods based on gradients and colour-segmentation are being actively researched today.

Another aspect worth exploring is robust estimation in the context of the pixel-based approach. The difference term $\mathcal{P}(u)$, used in the regularization approach, was set to the quadratic function. The principles of robust estimation applied to the block-based approach are general, and apply to the pixel-based approach as well. Why not replace the quadratic function with something offering a higher breakdown point as we did in the BM scheme? Of course, changing the difference term results in a different set of equations describing the optimal disparity function $(u, v)$, but it is worth trying.

This thesis has focused on disparity estimation and intermediate view reconstruction, for use in novel applications of stereoscopic video. The image reconstructions we have obtained, although not perfect, demonstrate that it is indeed possible to achieve high quality virtual views for stereoscopic video systems. Results we have presented recently stimulated interest and are very encouraging.

The technology presented herein is far from being realizable today in hardware, but with the ever-advancing nature of both hardware efficiency and consumer demand, the spectrum of applications for stereoscopic video is widening. Three-dimensional television, for example, is being considered as a next-generation medium, and it is fueling rapid developments of various relevant technologies. Although much of the attention today is on the compression of stereoscopic video, an equally active issue is stereoscopic systems with look-around capabilities. In this thesis, the role that intermediate view reconstruction has to play in this evolution has been defined, and interesting suggestions for its implementation have been presented.

# Bibliography

[1] A. Kopernik, R. Sand, and B. Choquet, "The future of three-dimensional tv," in *Signal Processing of HDTV, IV* (E. Dubois and L. Chiariglione, eds.), pp. 17–29, Elsevier Science Publishers B.V., 1993.

[2] T. Motoki, H. Isono, and I. Yuyama, "Present status of three-dimensional television research," *Proc. IEEE*, vol. 83, pp. 1009–1021, July 1995.

[3] L. F. Hodges and E. T. Davis, "Geometric considerations for stereoscopic virtual environments," *Presence*, vol. 2, no. 1, pp. 34–43, 1993.

[4] T. Naemura, M. Kaneko, and H. Harashima, "3-D segmentation of multi-view images based on disparity estimation," in *Proc. SPIE Visual Communications and Image Process.*, vol. 2727, pp. 1173–1184, Mar. 1996.

[5] L. Lipton, *The CrystalEyes Handbook*. StereoGraphics Corporation, 1991.

[6] A. Woods, T. Docherty, and R. Koch, "Image distortions in stereoscopic video systems," *Proceedings of the SPIE, Stereoscopic Displays and Applications IV*, vol. 1915, pp. 1–13, Feb. 1993.

[7] B. Choquet and J. Fournier, "Importance of opto geometrical adjustments for stereoscopic television," *2nd International Conference on 3D Media Technology*, Nov. 1992.

[8] D. Papadimitriou and T. Dennis, "Epipolar line estimation and rectification for stereo image pairs," *IEEE Trans. Image Process.*, vol. 5, pp. 672–676, Apr. 1996.

[9] I. Sexton, T. Bardsley, and A. Bhoopal, "Errors in depth," *International Workshop on Stereoscopic and 3-D Imaging, Santorini, Greece*, pp. 235–242, Sept. 1995.

[10] T. Naemura, M. Kaneko, and H. Harashima, "3-D object based coding of multi-view images," in *Proc. Picture Coding Symposium*, pp. 459–464, ??? 1992.

[11] N. Chang and A. Zakhor, "View generation for three-dimensional scenes from video sequences," *IEEE Trans. Image Process.*, vol. 6, pp. 584–598, Apr. 1997.

[12] J. Liu and R. Skerjanc, "Construction of intermediate pictures for a multiview 3D system," in *Proc. SPIE Stereoscopic Displays and Applications*, vol. 1669, pp. 10–19, Feb. 1992.

[13] R. Skerjanc and J. Liu, "A three camera approach for calculating disparity and synthesizing intermediate pictures," *Signal Process., Image Commun.*, vol. 4, pp. 55–64, 1991.

[14] M. Siegel, S. Sethuraman, J. S. McVeigh, and A. Jordan, "Compression and interpolation of 3D-stereoscopic and multi-view video," in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems IV*, vol. 3012, (San Jose, CA), pp. 227–238, Feb. 1997.

[15] S. B. Kang, "A survey of image-based rendering techniques," tech. rep., Cambridge Research Laboratory, Apr. 1997.

[16] M. Okutomi and T. Kanade, "A locally adaptive window for signal matching," *Intern. J. Comput. Vis.*, vol. 7, pp. 143–162, 1992.

[17] M. B. Slima, J. Konrad, and A. Barwicz, "Simple and effective stereo disparity estimation using sliding blocks and balanced filtering," *to appear in IEEE Trans. Circuits Syst. Video Technol.*, Dec. 1997.

[18] R. March, "Computation of stereo disparity using regularization," *Pattern Recognit. Lett.*, vol. 8, pp. 181–187, Oct. 1988.

[19] S. Malassiotis and M. Strintzis, "Joint motion/disparity estimation for stereo image sequences," in *Proc. SPIE Visual Communications and Image Process.*, vol. 2308, pp. 614–624, Sept. 1994.

[20] B. Chupeau, "A multiscale approach to the joint computation of motion and disparity: application to the synthesis of intermediate views," in *Proc. European Workshop on 3D-TV Signal Process.*, (Rome, Italy), Oct. 1993.

[21] E. Izquierdo and M. Ernst, "Motion/disparity analysis for 3-D video-conference applications," in *Int. Workshop on Stereoscopic and 3D Imaging*, (Santorini, Greece), pp. 180–186, Sept. 1995.

[22] A. Sethuraman, M. Siegel, and A. Jordan, "Segmentation based coding of stereoscopic image sequences," in *Proc. SPIE Digital Video Compression: Algorithms and Technologies*, vol. 2668, Jan. 1996.

[23] D. Tzovaras, N. Grammalidis, and M. Strintzis, "Object-based coding of stereo image sequences using joint 3-D motion/disparity segmentation," in *Proc. SPIE Visual Communications and Image Process.*, vol. 2501, pp. 1678–1689, May 1995.

[24] T. Aach and A. Kaup, "MAP-estimation of dense disparity-fields for stereoscopic images," in *Proc. 2-nd Singapore Intl. Conf. on Image Process.*, pp. 113–117, Sept. 1992.

[25] M. Ziegler, *Region-based analysis and coding of stereoscopic video.* PhD thesis, Technical University of Delft, 1997.

[26] J. Konrad, A. Golembiowski, and S. Coulombe, *ViDS Guide: INRS Format for Image Sequence Files.* INRS-Télécommunications, 3.0 ed., Aug. 1995.

[27] R. Franich and R. ter Horst, "Balance compensation for stereoscopic image sequence sequences," *International Organization for Standardization*, vol. Coding of Moving Pictures and Associated Audio Information - ISO/IEC JTC1/SC29/WG11 - MPEG96, Mar. 1996.

[28] R. Franich, *Disparity estimation in stereoscopic digital images.* PhD thesis, Technical University of Delft, 1996.

[29] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection.* New York: John Wiley & Sons, Inc., 1987.

[30] M. Black, *Robust incremental optical flow.* PhD thesis, Yale University, Sept. 1992.

[31] H. Sawhney and S. Ayer, "Compact representations of videos through dominant and multiple motion estimation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 814–830, Aug. 1996.

[32] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, pp. 1153–1160, Dec. 1981.

[33] E. Hendriks, S. Lanneau, A. de Ridder, and R. Nouta, "A hardware realization of a block-based disparity estimator on a cmos sea-of-gates semi-custom chip," *International Workshop on Stereoscopic and 3-D Imaging, Santorini, Greece*, pp. 305–310, Sept. 1995.