CORRESPONDENCE ESTIMATION AND INTERMEDIATE VIEW RECONSTRUCTION

Serdar Ince

Jan 2004

Boston University

Department of Electrical and Computer Engineering

Technical Report No. ECE-2004-01

BOSTON
UNIVERSITY

CORRESPONDENCE ESTIMATION AND INTERMEDIATE VIEW RECONSTRUCTION

Serdar Ince

Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
www.bu.edu/ece

Jan 2004

Technical Report No. ECE-2004-01

Summary

Stereoscopic vision is an important research area since 1861 when Wheatstone introduced the first stereoscopic device. After his basic device many technologies have been proposed to deliver 3D vision to audiences. Although most of these technologies require the viewer to wear special eyewear, new technologies are emerging that do not require glasses. These technologies, called autostereoscopic displays, are becoming very popular. However their ability is limited due to their demand of multiple high quality images. Common stereoscopic cameras shoot two images simultaneously which is not enough for these displays. Therefore, the problem to be solved is to generate intermediate virtual views using minimum number of real world images. The problem has two steps: correspondence estimation and view reconstruction. Correspondence estimation, which can be described as matching the same features in multiple images, is an ill-posed problem. Thus, many techniques have been proposed for correspondence estimation. In this report, basic methods used for this problem will be covered. The second step, intermediate view reconstruction, uses the information extracted in the first step, and works towards creating high quality views. Methods used for view reconstruction will also be covered in this report.

Contents

List of Figures

Chapter 1

Introduction

Visual aids are vital parts of our lives. One could not imagine a life without photographs or videos. Therefore, image processing is a huge scientific area where thousands of researchers are actively working. Although many of these researchers concentrate on two-dimensional (2D) image processing, there are many researchers in three dimensional (3D) vision research area as well. As a result of their work, in the future, we will be using more three dimensional visual aids than we use today. Nowadays, 3D devices that ordinary people are exposed to are limited to specially equipped movie theaters that require the audience to wear special kind of glasses. However, in the future, TVs that give us depth feeling without a need of any eyewear will be common in our homes.

It is actually very interesting that works on three dimensional images had begun long before photography was invented. It is noted that Leonardo da Vinci (1452-1519) studied the perception of depth [2]. His interest and appreciation for the third dimension can be found in his notes written 500 years ago. He writes *"...the main objective of painting is to show a raised body projecting from a plane surface. Whoever achieves this surpasses all others and should be considered most skilled within his profession."* [1].

However, the breakthrough happened in 1838 when physicist Charles Wheatstone (1802-1875) invented the first stereoscopic viewer. Wheatstone, a British scientist, explained the theory of stereoscopic vision in his address entitled *"Phenomena of Binocular Vision"* to the Royal Scottish Society of Arts in June 1838. The original speech can be found on the internet [6]. He constructed the stereoscopic device on the knowledge that left and right eyes view the same scene with a slightly different angle. This knowledge constructs the basics for stereoscopy.

After the invention of photography, Wheatstone also introduced the first photographic stereoscope in 1841.

Following Wheatstone, David Brewster (1781-1868) made significant contributions to the stereoscopic field. In 1849, he proposed his own stereoscope consisting of a pair of half lenses and an opening to a slot where the pair of images can be mounted side by side [8]. The new design was more compact than that of Wheatstone's. One year later he succeeded in interesting the French optics company Soleil and Duboscq and the

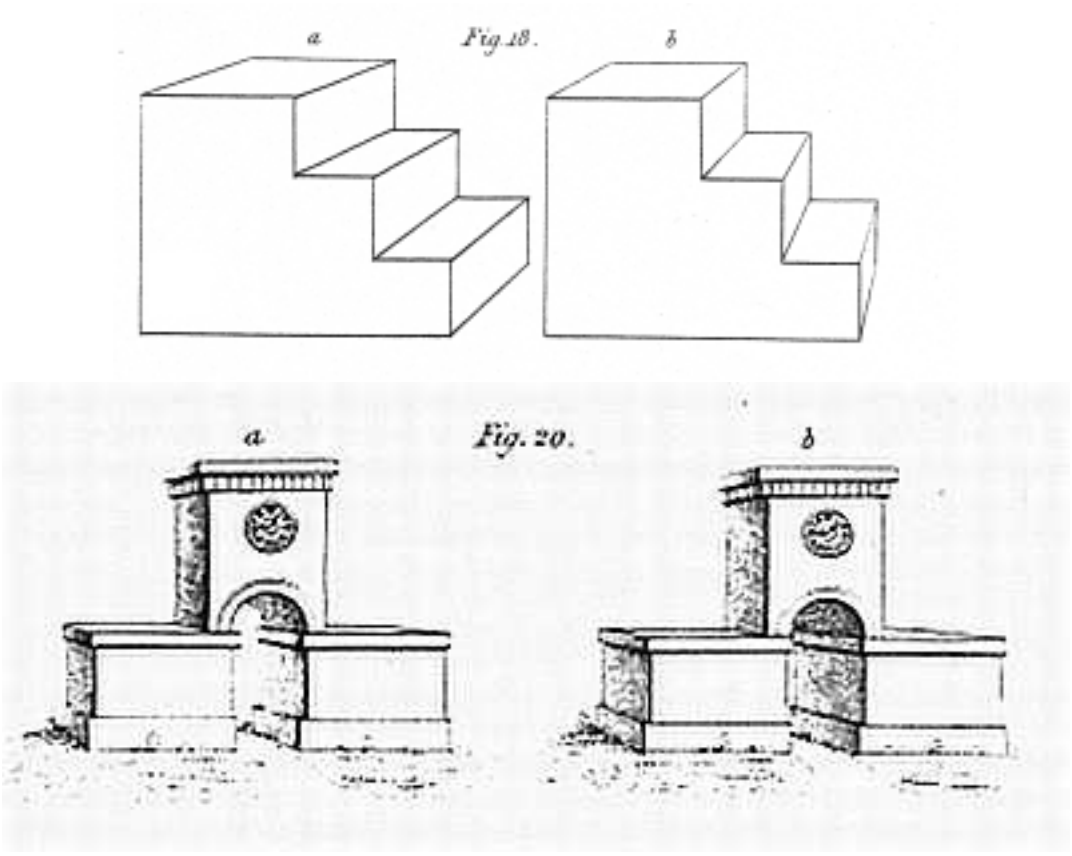

Figure 1.1: Wheatstone used hand drawings to demonstrate the function of his stereoscopic device

company started manufacturing the device [3].The company exhibited the stereoscope in an exhibit in London. Queen Victoria was very impressed by the device. Having caught the attention of the noble, it also drew the public's attention. As the result, within a three months period, a quarter of a million stereoscopes were sold in London and Paris.

Since the first stereoscopic device, many new techniques were introduced for stereoscopic displays. In this report, we will cover these technologies and we will mainly focus on stereovision related research topics.

Specifically, the report will start with the discussion of how humans perceive depth which will be followed by technologies used for stereoscopy. Also, applications of stereoscopy will be presented in this chapter.

Next, we will define correspondence estimation and intermediate view reconstruction (IVR). Applications of IVR and then correspondence estimation techniques along with comparisons will be presented. Next, IVR techniques will be discussed.

The report will conclude with a work plan and conclusions.

1.1 Definition of problem

The problem to be investigated is to create high-quality intermediate views using a number of cameras. For this purpose a number of techniques will be introduced to solve the following problems:

- Reliable estimation of depth using multiple cameras
- Transformation of the resulting depth maps to virtual viewpoints
- Reconstruction of virtual view at the new viewpoint

Following this phase, transmission of multiple views to users will be the main topic. The problems in this phase are:

- Evaluation of
  (a) transmission of all multiple views
  (b) transmission of a minimum number of views which are going to be used at the receiver end to create the extra views.
- If method (a) is preferred, compression techniques to reduce the redundancy should be investigated.
- If method (b) is preferred, optimization of IVR methods should be investigated.

1.2 Goals

A prototype system which renders required number of high-quality views from minimum number of cameras and then compresses them for fast transmission will be completed at the end of the project. Solutions to each problem mentioned in the previous section will act as milestones in the project.

Chapter 2

Perception of depth

How humans perceive depth is covered in this chapter. Monocular and binocular depth cues are introduced.

2.1 How do we perceive depth?

Depth is the relative distance of objects to the observer within a scene. Perception of depth is achieved by coordination of our eyes and brain. Human brain uses the depth cues, which can be defined as clues that help observer to find out the depth.

There are several types of depth cues that are used by human brain. They are mainly divided into two groups as monocular and binocular depth cues.

2.1.1 Monocular depth cues

Monocular depth cues don't require the use of two eyes and as the name implies, they can be viewed with one eye.

One of the basic monocular depth cues is *light and shade* [38]. Artists often use this cue in their works. Shadows give an idea about the shape of objects and help us to estimate the relative positions of points. Fig. 2.1 [38] shows this effect.

Another monocular depth cue is *interposition*. If an object is blocking the view of another object, then it is obvious that it is the closer to the viewer. For example, the gray square in Fig. 2.2 is closer to the viewer than the black one.

Yet another monocular depth cue is *relative size*. We know that objects that appear larger are closer to us. Therefore, if two objects are known to have the same size and if one of them is relatively bigger than the other, then we know that the bigger one is closer to us. The house on the left in Fig. 2.3 is closer to the viewer.

Aerial perspective is defined as the blurring of distant objects in the scene because of the haze and scattering of light in the atmosphere. The objects in Fig. 2.4 are same sized but the blurry one seems to be more distant to the viewer.

Textual gradient is another monocular depth cue [38] which is defined as gradual change in appearance of objects from coarse to fine. The objects that appear more

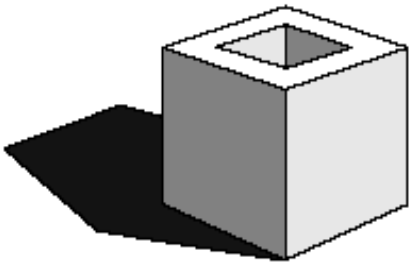

Figure 2.1: Shadows provide depth a cue

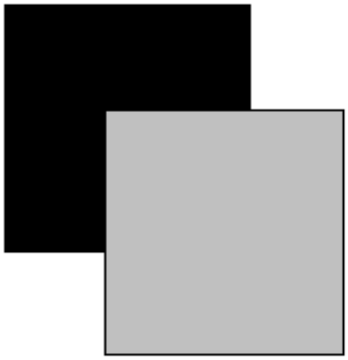

Figure 2.2: Interposition of objects provides a depth cue

distinct appear closer. Fig. 2.5 shows this effect. The lines in the lower side of the figure appear closer because of the textual gradient.

Motion parallax [38] depth cue can be noticed in case of continuous movement of objects. If two objects are known to have the same speed, the closer object appears to be moving faster than the distant object. This effect can be easily noticed when travelling in a car. As the car moves, hills in the landscape move very slowly, while the traffic signs on the road pass by very quickly.

Objects appear to be getting smaller as they get further away from the viewer. This effect is called *geometric perspective* [25]. Consider the example in Fig. 2.6. Point B seems to be closer than point A.

Figure 2.3: Relative sizes of objects provides a depth cue

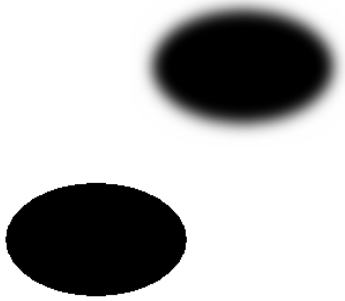

Figure 2.4: Blurriness of objects provides a depth cue

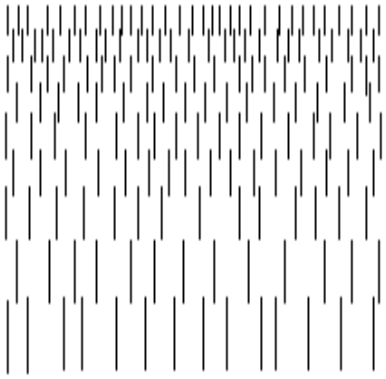

Figure 2.5: Textual gradient provides a depth cue

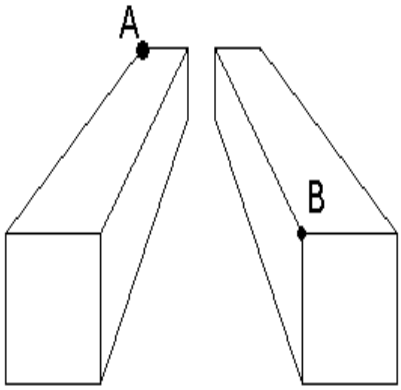

Figure 2.6: Geometric perspective provides a depth cue

2.1.2 Binocular depth cues

Unlike monocular ones, binocular depth cues are perceived by two eyes.

Human eyes are positioned at slightly different locations. The average interocular distance for humans is approximately 2.6 inches (6.5 cm) [27]. Due to this distance between eyes, the projections of the scene onto retinas of eyes are slightly different. The brain fuses these two slightly different images to give depth feeling. Fig. 2.7[1] shows the projection of a scene onto two cameras placed at interocular distance. If examined closely, one can notice the differences. For example, the farthest car that is in front of the bushes is more visible in the left image, while most parts of the car are occluded in the right image. Other differences can be noticed especially at object boundaries.

It is also easy to notice that positions of objects are slightly different in the two images. In order to understand this effect, consider this easy experiment: While looking at a stationary scene, close and open one eye at a time. It is easy to notice the differences.

Ignoring occlusions, a topic which will be covered later, all points in the left image are also present in the right image with a displacement due to the projection of scene. This displacement of a point in two images is called *disparity*.

The disparity is larger for objects closer to the viewer[2]. One can do the previous experiment to observe this fact, too. Put your thumb nearly 10 cm far away from your eyes. While looking at a distant object open and close one of your eyes. It is easy to notice that your thumb 'moves' quite a bit, while the distant object 'moves' little.

Using this basic information and camera geometry, the depth of each pixel in the scene can be constructed using stereo images. Depth value for each pixel can be

[1] Used with permission of www.stereovision.net

[2] It should be noted that we are considering parallel camera configuration here. In case of toed-in configuration, this statement may be invalid. Camera configurations are covered in Chapter 5.

Figure 2.7: Image on the left shows the scene through left camera and image on the right shows it through right camera

regarded as a new image, which is called the *depth (or range) map*.

Chapter 3

Intermediate view reconstruction

One of the main problems of stereoscopic systems is that they don't offer the freedom of choosing the viewpoint. The observer should watch the scene at the position of cameras. When watching a stereoscopic image/video, if the viewer moves the scene stays the same, which is not realistic. In real life, if one moves his head, he can see around the objects. This is called *look-around* feeling which stereoscopic systems cannot deliver [25].

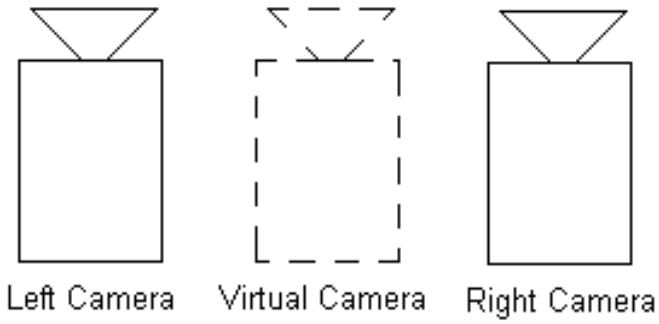

Figure 3.1: Intermediate view reconstruction

The solution to this problem is shooting the same scene at many different viewpoints and showing two of the views at a time to the user as he moves his head. Without any novel approach, using many cameras for recording the scene seems to be the easiest way. However, this kind of setup will be not only very expensive but also very bulky. Instead, we can use special techniques to create views at virtual viewpoints using a small number of real cameras. Creating virtual views using stereoscopic images is called *intermediate view reconstruction*[1]. In Fig. 3.1 we show such a setup. The scene is recorded using two cameras, the question is what the scene would look like from the position of the virtual camera. We will return to intermediate view reconstruction later.

[1]also called view synthesis

3.1 Distinction of terms

Since many terms such as stereoscopic, stereographic and 3D are used in literature, it is appropriate to begin by making distinction between them.

A stereographic image is a single image which contains meaningful and correct monocular depth cues. Therefore all the images that are captured using a camera are stereographic images [25].

Two stereographic images, one for left eye and one for right eye are called stereoscopic images or stereo-pair. If these images are properly delivered to viewever's eyes, he/she will perceive the depth of the scene.

The viewpoint[2] in stereoscopic images is fixed. The viewer sees through the viewpoint of the cameras. There is no freedom of choosing a viewpoint. The systems that let the users to determine the viewpoint for the scenes are called 3D systems [25].

[2]Here, we use the term *viewpoint* to describe the position of cameras.

Chapter 4

Stereoscopic display technologies

This chapter elaborates the benefits of stereoscopic display systems and technologies. In this chapter, we will discuss these technologies, with the emphasis on autostereoscopic displays.

4.1 Benefits and applications of stereoscopic display technologies

The first application area of stereoscopic vision is entertainment. Adding the third dimension to the current technology will add realism and will be more pleasing to audiences. IMAX ™ movie theaters are well-known among people for such an experience. Overcoming hardware and software problems, stereoscopic TVs can replace ordinary TVs in the future.

Businesses will benefit from stereoscopic displays, too. Companies can demonstrate their products in life-size using 3D displays. A customer in another country can examine the product without visiting the company or waiting for a sample. Similarly, virtual visits of homes, vacation sites can be made possible [34].

Stereoscopic vision can also be used in remote handling applications that are potentially hazardous to humans. Examples are underground works and radioactively contaminated areas. The field of medicine can use stereoscopy for surgeries.

All kind of simulators can be improved using stereoscopy. The trainee will be more aware of the situation with the enhanced display.

4.2 Current technology

Color-multiplexed (Anaglyph) displays: First anaglyphs were invented by Rollman in 1853. Although this technology is very old, it is still used due to its simplicity and little cost.

Anaglyph glasses have a red lens on left eye and a blue (sometimes cyan) lens on right eye. Anaglyph stereo images are produced in such a way that two images, a red

and blue image, are superimposed, one on top of the other. Since anaglyph glasses have colored lenses, left and right eyes see the red and blue images respectively. Since two different images are perceived by each eye, viewer can perceive the depth.

Crosstalk, which is defined as each eye seeing some part of the image of the other eye, is too high in this method. Also *color rivalry*, where homologous points are presented with non-matching colors, restricts the use of this method [55]. Many examples of anaglyph works and displays can be found on the internet.

Polarization-multiplexed displays: In this method, two views are projected through light polarizers onto the screen. These superimposed images are then separated by polarized (circular or linear) glasses worn by the user. The advantages of this method are little crosstalk and inexpensive and lightweight polarizing glasses [55]. An example polarized display product NuVision 17SX™ can be found at [4].

Time-multiplexed displays: This technique exploits the fact that human visual system can retain an image for some time. The left and right views are sequentially shown on the screen and lenses of the liquid crystal shutter glasses open and close in synchronization with the displayed images. When the left view is on the display, right lens occludes the right eye and vice versa. The operation of glasses is controlled by a infrared emitter placed close on the monitor. When the observer turns away from the screen, shutter glasses stop working and both lenses switch to open, so the viewer is not affected during normal viewing. Persistence of phosphors of CRT monitors may result in some crosstalk in this method [55]. An example product, CrystalEyes 3™ can be found at [7].

Autostereoscopic displays: While the techniques above require eyewear, autostereoscopic displays don't require eyewear. In this method, views are spatially multiplexed on a pixel-addressable screen and then separated using an optical layer that covers the display [34]. One example for the optical layer is lenticular sheet. SynthaGram™ 3D monitor, a product of StereoGraphics Corp., uses lenticular sheets for 3D viewing. This specific monitor is discussed in detail in the next section. More information on 3D display technologies can be found in [55].

4.3 SynthaGram 3-D display

The SynthaGram™ is an autostereoscopic monitor manufactured by StereoGraphics Corp. Most important features of SynthaGram are its capability to deliver 3D view without the need of eyewear and its capability to give look-around feeling.

SynthaGram™ monitors use lenticular sheet (sometimes called lens sheet or micro lens) which can be regarded as many miniature lenses arrayed on a flat sheet. This sheet covers the surface of a flat panel monitor. The screen and lenticular sheet are precisely aligned using Moir Interferometry[1]. The mapping of pixels on the monitor is predefined, a process called *interzigging* by StereoGraphics Corp.

[1]Moir Interferometry is an optical technique that uses lasers to measure displacements and strains on the surface of objects. Moir gives a full-field map of the strains. It has subwavelength displacement sensitivity[5].

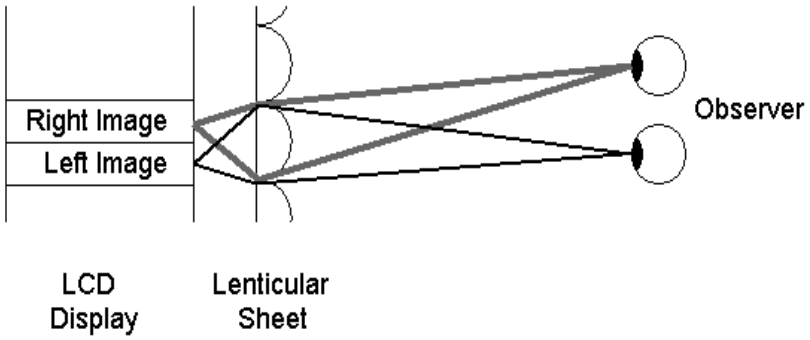

Figure 4.1: Lenticular sheet

The main idea of using lenticular sheet is summarized in Fig. 4.1. The basic principle of delivering different images to each eye is achieved using optical lenses of the lenticular sheet. One can think of this configuration as of eyewear, that the user should wear, is carried on the monitor itself.

As already mentioned, the SynthaGram^{TM} is capable of giving look-around feeling. This is achieved by creating a number of view zones using nine images. These nine images of the scene should be taken by evenly spaced cameras. Since there are many images shown on the monitor, the user will be able to see different perspectives as he moves his head. To sum up, in a single view zone, user can experience a look-around feeling. If the user enters another view zone, same perspective progression is repeated. There are five view zones which are created using nine images. Using nine images wass selected after experiments of developers. Although nine images are required in order to have the best quality, fewer images can be used, but results will be degraded, since there will be distinct jumps between perspectives.

It should also be noted that another advantage of having multiple view zones is that multiple users can see the same 3-D image effortlessly.

Since there are nine images for the display, there are nine times more pixels than in a single image, therefore they should be processed before displayed on the monitor. This processing is called *interzigging*. The first step of interzigging is downsampling of images to reduce the number of pixels. Since there are nine images, downsampling by three in both x and y directions is appropriate. It should also be noted that this downsampling process obviously decreases the image quality, which is a drawback of the SynthaGram^{TM}. Proper prefiltering stage needs to be applied to prevent aliasing. Special anti-aliasing filters for SynthaGram^{TM} monitors are discussed in [35] .

The next step is choosing the correct pixel from the source image and assigning it to its new location in the interzigged image. In an early version of such displays manufactured by Phillips, the following formula is used to choose pixels and assign them on the monitor [67].

$$N = \frac{(k + k_{offset} - 3\,l \tan\alpha)\, mod\, X}{X} N_{tot} \tag{4.1}$$

where

k,l: pixel location

N-tot : total number of views

α: angle of slanted lenticular

X: view per lens

k_{offset}: a variable to accommodate horizontal shift of the lenticular lens array (usually zero)

Image #1 (Leftmost Image)	Image #2	Image #3
Image #4	Image #5	Image #6
Image #7	Image #8	Image #9 (Rightmost Image)

Figure 4.2: Nine-tile format

After the interzigging operation there will be a single image to view on the SynthaGram™with the same dimensions as the original images.

However, a problem arises after interzigging. Since the new image is a 'shuffled' image, compression significantly degrades the quality due to blocking artifacts. Blocking artifact is a problem in block based-lossy compression standards, e.g., JPEG. Due to the quantization step in compression algorithms, it is easy to see the boundary changes between blocks at high quantization levels. Although this blocking effect can be tolerated to some level in ordinary images, it results an unpleasant experience on SynthaGram™monitors, because blocking artifact causes texture *and* depth distortion.

In order to overcome the compression and transmission difficulty, developers introduced the nine-tile format, which is basically downsampling the images by three in each direction and concatenating them to create a single image with the same size of

the original images. The images should be properly prefiltered before downsampling to avoid aliasing.

Another reason for the nine-tile format is that for different monitor resolutions different interzigging algorithms are required. Therefore original images are required to create an interzigged image for a different monitor. Thus it is an appropriate way to store the images.

Chapter 5

3-D geometry

In this chapter, we will define basic concepts that we frequently use in 3-D processing such as camera model, projective and epipolar geometry.

5.1 Pinhole camera model

Since we work on images taken by cameras, it is essential that we have a camera model to extract quantitative measures. Therefore, we will first define the basic model used in the literature which is the *pinhole camera model*. Although being very simple, this model accurately models the geometry and optics of most modern cameras [24].

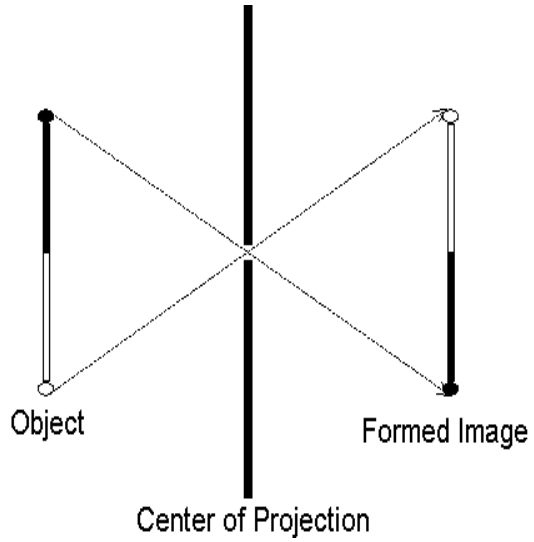

Figure 5.1: Pinhole camera model

The image formation process in a pinhole camera model is illustrated in Fig. 5.1. All of the light rays departing from the object pass through a small hole, called *center of projection* or *optical center*, and forms the inverted image on the screen. Now we

will elaborate on the geometry, and derive the equations that describe projective geometry.

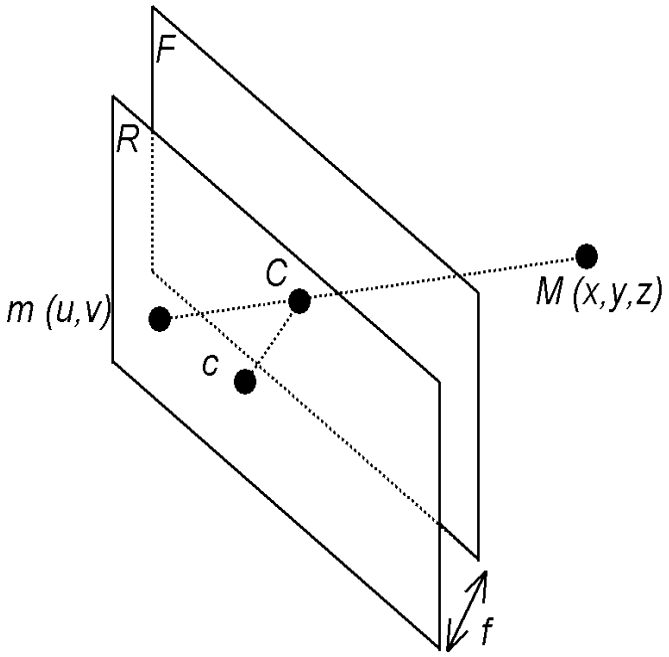

Figure 5.2: Projection of a point in 3D world onto image plane

Consider Fig. 5.2 which shows two planes: Plane $\boldsymbol{F}$, the focal plane, is where the center of projection resides, and plane $\boldsymbol{R}$, the retinal plane, is where points in three-dimensional space are projected onto. The distance between plane $\boldsymbol{F}$ and $\boldsymbol{R}$ is called *focal length*. Now let us define two coordinate systems, one for the three dimensional space (world coordinates) (C, x, y, z) which is centered at C, and one for the retinal plane (c, u, v) which is centered at c.

Now let's study the point $\boldsymbol{M}$ with coordinates (x,y,z) in three-dimensional space. $\boldsymbol{M}$ is going to be projected onto $\boldsymbol{m}$ on the retinal plane $\boldsymbol{R}$ through the point $\boldsymbol{C}$ on the focal plane, thus resulting in point $\boldsymbol{m}$.

Using basic geometric relation of similar triangles, the following relation can be written

$$-\frac{f}{z} = \frac{u}{x} = \frac{v}{y} \tag{5.1}$$

Rewriting the equations we get

$$\begin{bmatrix} u \\ v \\ z \end{bmatrix} = \begin{bmatrix} -f/z & 0 & 0 & 0 \\ 0 & -f/z & 0 & 0 \\ 0 & 0 & 1/z & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5.2}$$

where $z \neq 0$

Next, we define an additional component S in order to convert the above relationship to linear, so that we can define an image point with a triplet *(U,V,S)* where

$$u = U/S$$

$$v = V/S$$

and

$$S \neq 0$$

This additional component comes from the idea that (U, V, S), $(u, v, 1)$ and (u, v) should represent the same point.

Finally combining previous equations, we derive the following equation.

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5.3}$$

or

$$\vec{\mathbf{m}} = \vec{\mathbf{P}}\vec{\mathbf{M}} \tag{5.4}$$

where

$$\vec{\mathbf{m}} = [U\ V\ S]^T \; and \; \vec{\mathbf{M}} = [x\ y\ z\ t]^T.$$

To sum up, in (5.4) we project a point in world coordinates to the camera plane with the help of a linear equation.

5.2 Stereo camera setups

There are two primary camera configurations in the literature.

The *parallel camera configuration* uses two cameras with parallel optical axes. This configuration is frequently used due to its simplicity in both setup and mathematical derivations. Since the optical axes are parallel, they will converge at infinity and infinity will projected on the screen. This may result in excessive screen parallax on the screen and can cause viewing difficulties for the observer. Another problem is

that depending on the distance between the cameras the common field of view may be small. This creates a problem at the correspondence matching stage.

The second configuration is *toed-in camera* or *converging camera configuration* where cameras are a little rotated towards each other, so that the optical axes of cameras converge at some point other than infinity. The advantage is that the common field of view is increased. However, this configuration introduces opposing keystone distortions in the stereo pair. Keystone distortion can be described as distortion that projects a rectangular figure as a trapezoid in the picture.

A comparison of these two configurations along with resulting distortions is discussed in [68].

5.3 Camera geometry

We will now define geometric relations for the camera setups mentioned in the previous section using the pinhole camera model.

5.3.1 Parallel cameras

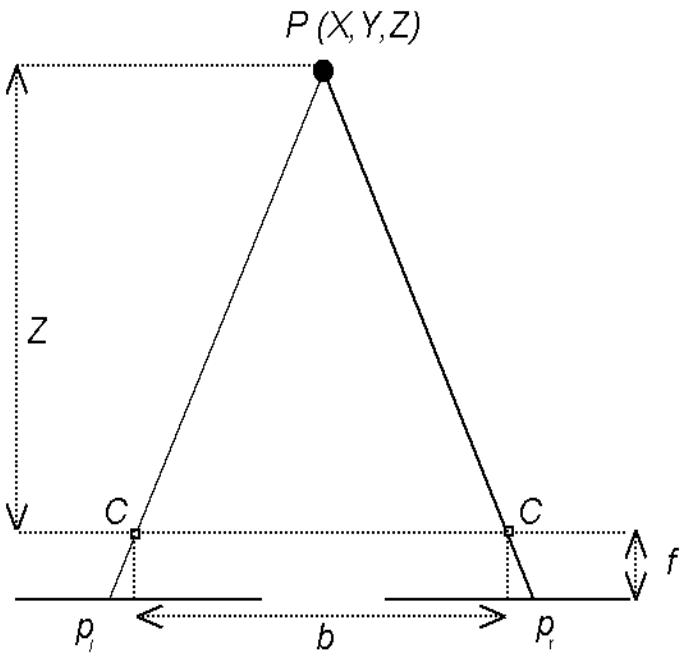

Figure 5.3: Projection of a point onto parallel cameras

The parallel camera configuration is illustrated in Fig. 5.3. Consider a point P in 3D world with coordinates (X, Y, Z) and its projection onto two points p_l and p_r on image sensors of the left and right cameras. Let the coordinates of points p_l and

p_r be (x_l, y_l) and (x_r, y_r), respectively. The horizontal distance b between the two cameras' optical centers is called the *baseline distance*. Finally, assuming that two identical cameras are used, let the focal length of the cameras be f.

Using the perspective transformation the following relations [25] can be written

$$x_l = f\frac{X + b/2}{f - Z} \tag{5.5}$$

$$y_l = f\frac{Y}{f - Z} \tag{5.6}$$

$$x_r = f\frac{X - b/2}{f - Z} \tag{5.7}$$

$$y_r = f\frac{Y}{f - Z} \tag{5.8}$$

It is obvious that y_l and y_r are identical. This is the advantage of the parallel camera setup. *Disparity* d is mathematically defined as

$$d = x_r - x_l \tag{5.9}$$

The Z component of point P can be also computed using similarity of triangles (P, p_l, p_r) and (P, C, C).

$$\frac{Z}{Z + f} = \frac{b}{b + d} \tag{5.10}$$

$$Z = \frac{b\,f}{d} \tag{5.11}$$

5.3.2 Converging cameras

In converging camera setup, cameras are a little rotated towards each other at angle β so that their optical axes intersect at a convergence point. For the setup given in Fig. 5.4 the convergence point is at

$$Z_{convergence} = \frac{b}{2\tan\beta} \tag{5.12}$$

The projection of a 3D point on convergenging cameras is more complicated than the parallel cameras case. In addition to steps in the parallel cameras case, there is also a rotation in projection. Using derivations given in [25] we can show the projection step by step

i) Translation

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} X + b/2 \\ Y \\ Z \end{bmatrix} \tag{5.13}$$

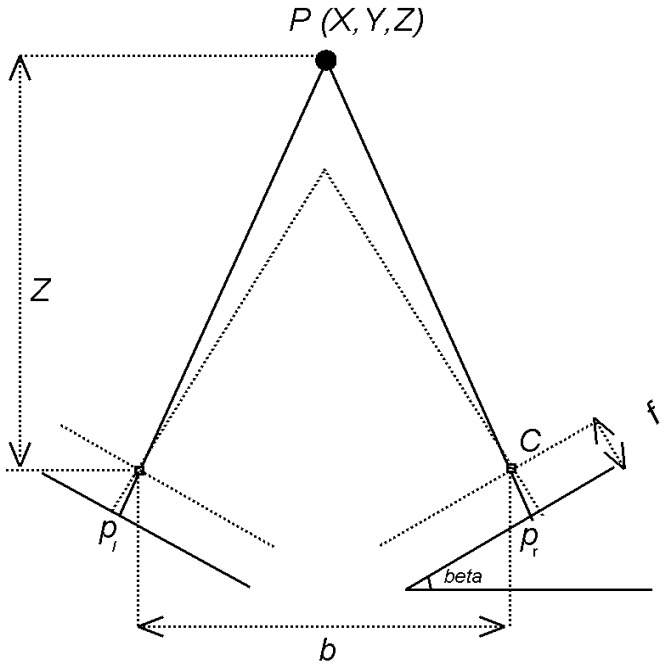

Figure 5.4: Projection of a point onto converging cameras

ii) Rotation around the y axis by angle of β

$$\begin{bmatrix} X + b/2 \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} \cos(\beta)(X + b/2) - \sin(\beta)Z \\ Y \\ \sin(\beta)(X + b/2) + \cos(\beta)Z \end{bmatrix} \tag{5.14}$$

iii) Perspective projection

$$x_l = f\frac{\cos(\beta)(X + b/2) - \sin(\beta)Z}{f - \sin(\beta)(X + b/2) - \cos(\beta)Z} \tag{5.15}$$

$$y_l = f\frac{Y}{f - \sin(\beta)(X + b/2) - \cos(\beta)Z} \tag{5.16}$$

$$x_r = f\frac{\cos(\beta)(X - b/2) + \sin(\beta)Z}{f + \sin(\beta)(X + b/2) - \cos(\beta)Z} \tag{5.17}$$

$$y_r = f\frac{Y}{f + \sin(\beta)(X - b/2) - \cos(\beta)Z} \tag{5.18}$$

Using these equations we can determine the horizontal disparity as

$$d = x_r - x_l = -2f\frac{N}{D} \tag{5.19}$$

where

$$N = 4\sin(\beta)\cos(\beta)Z^2 + \left[2b - 4f\sin(\beta) - 4b\cos^2(\beta)\right] Z$$
$$b\cos(\beta)\left[2f - b\sin(\beta)\right] + 4\sin(\beta)\cos(\beta)X^2 \tag{5.20}$$

and

$$D = 4\cos^2(\beta)Z^2 + 4\cos(\beta) - \left[b * \sin(\beta) - 2f\right] Z$$
$$\left[4f^2 - 4fb\sin(\beta) + b^2\sin^2(\beta)\right] - 4\sin^2(\beta)X^2 \tag{5.21}$$

For $\beta = 0$ the converging camera setup becomes equivalent to parallel camera setup.

5.3.3 Epipolar geometry

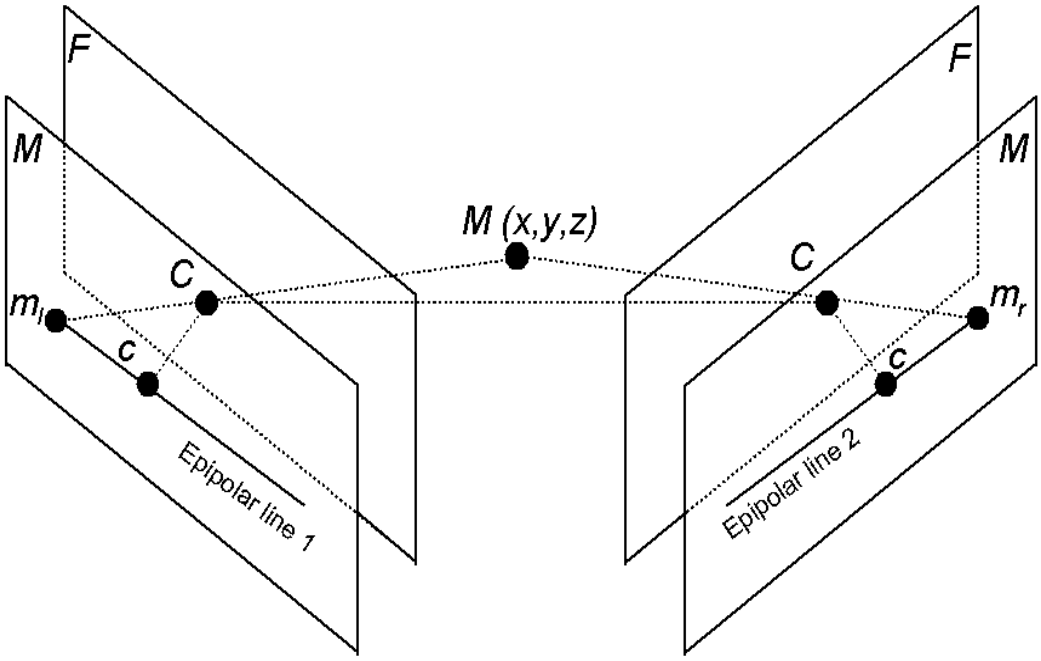

Figure 5.5: Epipolar geometry

Epipolar geometry introduces a very powerful restriction in correspondence estimation. Epipolar constraint was first introduced in [40].

In order to understand the epipolar constraint better, consider Fig. 5.5. It can be noticed that all possible physical points M that may have produced m_l are the infinite half line $\langle C, M\rangle$. As a direct consequence, all possible matches m_r of m_l on the right retinal plane are formed by this half-line through the second camera. This

image is a half-line *epipolar line 2* going through m_r which is the intersection of the line that connects the center of projections with the right retinal plane.

Using the latter observation, the epipolar constraint states that for a given m_l on left retinal plane, all its possible matches on the right plane lie on a line. After introducing this constraint, the matching process reduces to one dimension from two.

Chapter 6

Correspondence estimation techniques

In previous chapter, we have given the basics for stereo geometry which are used to solve stereo vision problems. The biggest obstacle for this problem is the *correspondence estimation*. Correspondence estimation can be described as finding the corresponding token (line, pixel, object etc.) in second image using a given token in the first image. By nature, correspondence estimation is *ill-posed* which means that the problem has no exact or unique solution or is overly sensitive to the date.

There are mainly two constraints introduced to make the problem simpler. The first constraint, *constant image brightness constraint* (CIB) [28], is introduced in order to be able to relate luminance values and correspondence. The constant image brightness constraint states that corresponding pixels in images should have equal luminance values. However, even this strong constraint is not enough for solving the problem. Consider an image which contains a contour whose points have the same luminance, then pick a point on the contour. Since all points have the same luminance values, what is the corresponding point in the other image? Thus, CIB is not enough to convert the problem to a well-posed one [58]. The second constraint is epipolar constraint which restricts the search range for a corresponding point in one image to the epipolar line in the other image.

Another problem in correspondence estimation is *occlusion*. As illustrated in Fig. 6.1, occlusion problem arises when a point is visible through one camera but not the other one. In such cases, correspondence estimation becomes impossible.

Yet another problem can be noticed from programming perspective. As opposed to motion estimation in video, the possible range of vectors in disparity estimation is larger, therefore there are more candidates to investigate which obviously requires very high computational effort [72]. Of course, in case of the epipolar constraint, number of calculations significantly decreases, however in real-world images there is almost always a vertical component of disparity vectors.

A technique called *rectification* [51] is used to 'adjust' images so that images are re-projected onto planes corresponding to parallel camera setup. In other words,

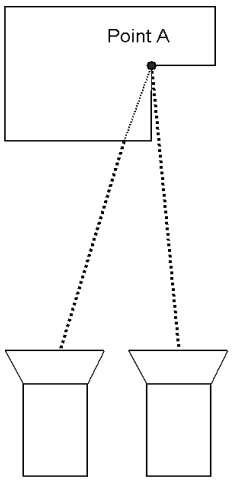

Figure 6.1: Point A is visible on right camera but not visible on the left camera

images are aligned in such a way that disparities in vertical direction are cancelled after rectification step. A free rectification toolbox for MATLAB can be found in [16].

Although rectification methods help reduce the computational load, they manipulate the original images, therefore may yield problems during intermediate view reconstruction. Specifically, during the process of rectification, images are resampled. Some of the data can be lost during rectification, especially details in small areas. The loss of data may degrade the quality of resulting intermediate views.

Now we will discuss methods used in correspondence estimation. Numerous methods have been proposed in literature:

Block matching algorithms: These techniques match rectangular blocks of pixels in images [39, 63, 69, 10, 23]. For a block B_{left} in the left image, a block B_{right} is searched in the right image in a predefined search window. The resemblance is evaluated using some cost function, for example Sum of Absolute Differences. Once the best match is found, same correspondence vector is assigned to all points in the block. Therefore this method a piecewise constant disparity map.

The advantage of this method is its simplicity. Many efficient block matching techniques are available in video compression. Hardware implementations are also available.

This method yields very good results in large textured areas that are displaced uniformly [58]. Results can be enhanced using different block sizes, however large blocks tend to work better due to carrying more features to match.

Although being very simple, this method cannot fully describe disparities that yield constant depths. Another problem is the assumption that states that all points in the block should have experienced the same displacement, therefore they should

have equal depth values. However, this assumption fails at object boundaries. The depth suddenly changes at object boundary, however since it is assumed that all points in a block should have the same depth, the resulting depth map will contain incorrect values if there is an object boundary in the block.

One solution for this problem is changing block size in problematic areas [44]. In potential boundary areas, the block size can be reduced to accommodate the sudden change in the depth. Although this works up to some level, detection of problematic areas and implementation are difficult.

Block matching is computationally expensive especially for large disparity values since a cost function is evaluated at every candidate point.

Feature matching: This method tries to match a specific feature between the images. The feature may be determined automatically or by the user. This can be a line, an edge or any other specific feature in the scene [15, 17, 64, 37].

The advantage of feature matching is that features used are usually more reliable than intensities when CIB constraint does not hold. However, feature matching may face problems if the feature extraction process yields missing or false features [49].

Although feature matching techniques can compute reliable and accurate correspondences, they compute sparse maps. No disparity information is computed for non-feature points.

Optical Flow: All previous methods work locally on the images, therefore resulting disparity map is sometimes not reliable. Moreover, in case of feature matching a sparse disparity map is computer. However a dense depth map which requires disparity vectors calculated for each pixel independently is required for applications such as intermediate view synthesis. Optical flow, proposed by Horn and Schunck in [29], is a seminal method that started the work towards calculating the motion globally.

Optical flow is the apparent motion of a brightness pattern. In theory, the optical flow is equal to motion field[1]. Let $E(x, y, t)$ be the brightness at time t at the image point (x, y), and $u(x, y)$ and $v(x, y)$ be the horizontal and vertical components of the optical flow at that image point. Using the constant image brightness constraint, we can write the following equality at $t + \triangle t$ where $\triangle t$ is a small time interval

$$E(x, y, t) = E(x + u\triangle t, y + v\triangle t, t + \triangle t) \tag{6.1}$$

Expanding the equation using Taylor series and making some approximations, we end up with the following equation which is called the *optical flow constraint equation.*

$$E_x u + E_y v + E_t = 0. \tag{6.2}$$

where

$$E_x = \frac{\partial E}{\partial x}, \quad E_y = \frac{\partial E}{\partial y}, \quad E_t = \frac{\partial E}{\partial t}$$

and

[1]Practically, this condition is not always satisfied in some cases such as rotation of an object.

$$u = \frac{dx}{dt}, \quad v = \frac{dy}{dt}$$

Working solely with 6.2 will yield an irregular optical flow map. However, since the optical flow will be very similar on the average, it can be smoothed by introducing additional constraints. Assuming the rigid body motion, we can smooth most parts of the image by minimizing the departure of velocities u and v from the smoothness. Specifically, we can minimize

$$e_s = \int\int ((u_x^2 + u_y^2) + (v_x^2 + v_y^2))\, dx\, dy. \tag{6.3}$$

After combining (6.2) and (6.3) and introducing a smoothness parameter λ, we minimize the following integral

$$\int\int ((u_x^2 + u_y^2) + (v_x^2 + v_y^2) + \lambda\, (E_x u + E_y v + E_t)^2\, dx\, dy \tag{6.4}$$

The corresponding Euler equations are

$$\nabla^2 u = \lambda(E_x u + E_y v + E_t)E_x \tag{6.5}$$

$$\nabla^2 v = \lambda(E_x u + E_y v + E_t)E_y \tag{6.6}$$

where

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

.

Although smoothing the optical flow using additional constraints yields better estimates, it also creates a major problem at object boundaries. There will be discontinuities at the object boundaries where the motion field (or the depth) suddenly changes. Therefore, regularization algorithms that are used to smooth the optical flow should force smoothness inside a rigid body but should refrain from smoothing across the object boundaries. March discusses this problem in [47]. He proposes to use a mask for the optical flow equation, which prevents smoothing across the boundaries.

Recognizing an object boundary is an easy task for a human but it is very difficult to detect in computational implementations. An edge in the image may be a boundary between objects or a textural edge as well. Explicitly marking the boundaries and preventing the algorithms work at these points may solve the problem.

After Horn and Schunck, many optical flow methods have been proposed in literature: Lucas and Kanade [41], Nagel-Enkelmann [50] are two of the popular ones. For a detailed comparison of optical flow methods, refer to Barron *et al.*'s paper [14] which quantitively evaluates several methods. McCane *et al.* also compares optical flow methods in [48].

The main drawback of optical flow methods is that they cannot capture displacements more than one or two pixels due to the approximation in derivation. Therefore, hierarchical scheme implementations are required to compute large displacements. Initializating the method using coarse displacements may be another solution.

Another drawback of this method is that the derivatives given above are numerically approximated. This requires a local spatiotemporal linearity of luminance which is not always the case for image sequences with large motion [58].

Phase-based methods: Another approach for correspondence estimation is based on phase of the Fourier transform of two images [31, 43, 42].

The idea behind phase-based methods is to compute the disparity as the spatial shift from the local phase differences in Fourier domain [42]. Usually both images are convolved with a filter and local phase is extracted from the result. Since the local shift is approximately proportional to the local phase difference, a disparity value can be estimated at each point. The filter usually exploited is Gabor filter. An FFT-based disparity estimation is also proposed in [11]. For a comparison of some of the phase-based disparity estimation techniques refer to [22].

Other methods: Since the problem is ill-posed, many other approaches for disparity estimation can be found in the literature. Although we will not give details of these methods, we will give some references to these work.

One approach is simultaneous estimation of correspondence fields combined with object segmentation using Bayesian methods [20, 36].

Other approaches that use partial differential equation (PDE) based methods can be found in [56, 12, 13, 66]. The basic idea of PDE based methods is to represent the image as the level set of a higher dimensional hypersurface. Using PDEs in image analysis helps to model the images in a continuous domain, which leads to grid-independence and isotropy [59]. Another advantage of variational and PDE methods is that they allow to directly handle and process visually important geometric features such as gradients, curvatures. They can also easily simulate dynamic processes such as nonlinear diffusion [19]. Moreover, the PDE approach can achieve accurate and stable solutions with the help of available research on numerical analysis [18].

Moreover, A Maximum A Posteriori (MAP) estimation technique can be found in [9].

Chapter 7

Intermediate view reconstruction techniques

In previous chapter, we presented some techniques to compute the correspondences between images. Now we are going to present techniques that utilize these computed correspondences to construct intermediate views.

For simplicity let's assume the number of cameras is two and the cameras are perfectly parallel. Similar approaches can be used to extend the method for more cameras.

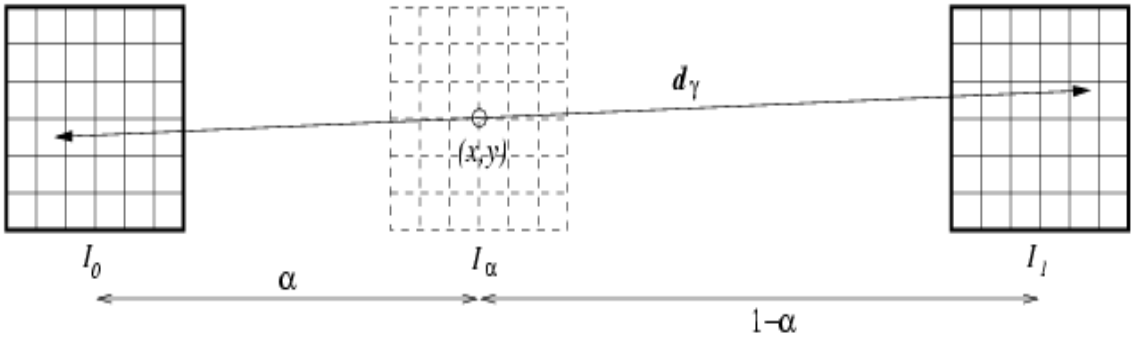

Figure 7.1: Intermediate view reconstruction pivoting on the intermediate view

We have two images $I_{left}(x, y)$, $I_{right}(x, y)$ and the correspondence (disparity) map $d(x, y)$ and we would like to reconstruct the intermediate view $I_\alpha(x, y)$ at relative position α, where $0 \leq \alpha \leq 1$.

In order to compute the intensity value of a point (x, y) on I_α, we pivot on intermediate view plane at (x, y) such that the disparity vector $d(x, y)$ passes through (x, y). Then, we have three points of interest[1]:

$$I_{left}(x - \alpha\, d(x, y), y)$$

[1] assuming vertical disparities are zero

$$I_{right}(x + (1 - \alpha\ )d(x, y), y)$$

$$I_{\alpha}(x, y)$$

The basic approach to calculate the intensity of $I_{\alpha}(x, y)$ is weighted averaging. Therefore we can write the following equation

$$I_{\alpha}(x, y) = (1 - \alpha)\, I_{left}(x - \alpha\, d(x, y), y) + \alpha\, I_{right}(x + (1 - \alpha\ )d(x, y), y) \qquad (7.1)$$

Since we pivoted on the intermediate view, intersections of a disparity vector with left and right images do not, in general, belong to their sampling lattices. These intensities should be computed using interpolation techniques such as bilinear interpolation or bicubic spline.

Instead of pivoting on an intermediate view, we can also pivot on left and right views, however in this case, most of the points reconstructed on intermediate view will not belong to the sampling lattice. The resulting irregularly-spaced points can be projected onto orthogonal lattice using such techniques as Projection onto Convex Sets (POCS) [65].

There are a few methods that utilize the approach described by (7.1). The basic method is block-based reconstruction [33]. One disparity vector is assigned per block and all points in the block are reconstructed using (7.1).

As already mentioned, the assumption that all points in a block should have the same disparity does not always hold. Therefore, it is appropriate to change the block size when required. Quadtree block matching [45] technique first calculates the disparity values for larger blocks and then reduces the block size at possible boundary locations. After estimation of disparity field, intermediate views are reconstructed using (7.1) again.

Linear interpolation methods that use equation (7.1) tend to result in blurry intermediate views. Mansouri and Konrad proposed a winner-take-all approach to overcome this problem [46]. In their approach, the intermediate view is reconstructed by tilings from either left or right images. In other words, every block in the intermediate view is equal to the corresponding disparity-compensated block in either the left or right image. Although this method decreases the blurr in reconstructed image, it also introduces a patchiness effect when left and right views have significant differences in intensities.

Another method by Scharstein computes two intermediate views at the same point by forward mapping (disparity compensating the images using disparity vectors) both left and right views [61]. The algorithm computes the disparity field using an intensity gradient method described in [60]. Then, both left and right images are forward mapped to intermediate location. Obviously there will be *holes* in forward mapped images since a point may be overdefined (i.e., more than two points in original image can fall on the same location in intermediate view) or the new point is not visible in both of the views. The next step is combining these two intermediate views. Weighted averaging is one of the methods to blend the images. Also, holes can be filled using information from one of the images. However, holes may still exist in the

intermediate view. At this point, Scharstein uses texture synthesis methods. This method being simple, suffers from the forward mapping part. Filling the holes using texture synthesis methods may be a problem for reconstructed views. Especially areas with details may not be reconstructed properly.

Redert *et al.* introduce a method which can reconstruct views at *non* intermediate positions [57]. Their motivation is to overcome the restriction that fixes the new viewpoint between the cameras. They compute correspondences using dynamic programming and working on blocks of 4x4 points. Then they compute the intermediate view at center point between the two cameras. For the rest of the algorithm, they use this center view and a single disparity map D (either right-to-center or center-to-left). They relate the center view to any view at position (S_x, S_y, S_z) with the following formula

$$\begin{bmatrix} X_v \\ Y_v \end{bmatrix} = \frac{S_{zoom}}{1 - S_z \frac{H_c}{f} D} \begin{bmatrix} X - S_x D \\ Y - S_y D R \end{bmatrix} \tag{7.2}$$

where H_c and V_c are sizes of the image in x and y direction respectively, $R = H_c/V_c$ the pixel ratio of the camera and f focal length of the camera.

Image rendering using equation (7.2) also suffers from problems like in Scharstein's case. Pixels in virtual image may be overdefined or remain undefined. In case of overdefinition authors select the point that is closest to virtual camera. For undefined case they use linear interpolation using the neighboring points to fill the holes. It should be noted that the method is successful in the sense of simplicity since it uses a single image and a single disparity map for any intermediate view.

Havaldar *et al.* proposed a view synthesis technique in [26] which uses projective invariants. They do not require knowledge of the camera positions. An invariant, defined with respect to a transformation T, is a property which remains unchanged under transformation T. For example, parallel lines always map to parallel line under orthographic projection. Also, the ratio of line segments with respect to each other named *cross ratio* remains unchanged. Their method starts by extracting feature points which are usually corners and faces. Next, these features are matched using correspondence estimation methods. Given any corresponding points on two images, the prediction of the point in new image is accomplished using cross ratio of four points in two images. Since for every point whose cross ratio is computed, the location of the point in new image can be calculated, these points are linked to form wireframes in the virtual view. After this step, the boundaries of objects in the scene are reconstructed. The final step is texture mapping which uses back projection using the computed projective transform.

Utilizing more than two cameras for view synthesis is also discussed in the literature. The advantage is that occluded parts may be better defined using additional views. Park and Inoue proposed an arbitrary view generation algorithm using five cameras [53]. Their camera system consists of a center camera and a total of four cameras to each direction of above, lower, left and right, separated by the same distance. The center camera is of special importance since all of the correspondence

estimations will include this camera.

In their algorithm, they exploit a hierarchical correspondence estimation algorithm given in [54]. This estimation method assumes that at least one of the two cameras located symmetrically (i.e., either left or right or upper or lower) is not influenced by occlusion, therefore gives correct matching data. At each step, they compare cost values of symmetric cameras and discard the larger one. This is equivalent to choosing the most proper configuration consisting of three cameras. Following this correspondence estimation step, they construct a depth map for the central camera using the disparity values.

In the next step, they relate the points (x, y) in central camera with the points (x', y') of a virtual camera at position (S_x, S_y, S_z) with the following equations:

$$x' = \frac{1}{1 - S_z/D(x,y)} \left(x - \frac{f\, S_x}{D(x,y)} \right) \tag{7.3}$$

$$y' = \frac{1}{1 - S_z/D(x,y)} \left(x - \frac{f\, S_y}{D(x,y)} \right) \tag{7.4}$$

where $D(x, y)$ is the depth map and f is the focal length.

By using (7.3) and (7.4), they forward map the depth map of central camera to that of the virtual camera. However, the problem of having overdefined or undefined points again arises due to the forward mapping. For overdefined points, they choose the smallest value among the multiple depth values. In order to fill the holes (i.e. undefined points) in transformed depth map, they use the concept of *observable viewpoint (OVP)*. OVP can be defined as the direction of mapping. For example, if the camera is moved to the right, the new image on the new viewpoint will contain some information that cannot be observed from the original viewpoint and all of the new points will be on the right of objects, because the camera is moved to the right. In summary, the OVP is the direction in which an uncovered area appears. Thus, they fill the empty points with the maximum depth value of two endpoints along the direction of OVP. After these steps they have a fully defined depth map at new viewpoint.

The reasons for choosing smallest depth value for overdefined points but choosing the maximum depth value for undefined points are not explicitly mentioned and seem to contradict each other.

The final step in the algorithm is texture mapping. This is easily done at points that can be observed from both primary and virtual cameras. For uncovered areas, they map a color value from one or two of the secondary images using the depth value guessed in the previous step. If none of the projection methods succeeds to fill the empty region, they interpolate the holes by weighted spreading of colors from neighboring pixels.

Finally, after reconstructing the virtual view, they change the orientation and scale of the image if desired. They employ 2D geometric transformations given in [52].

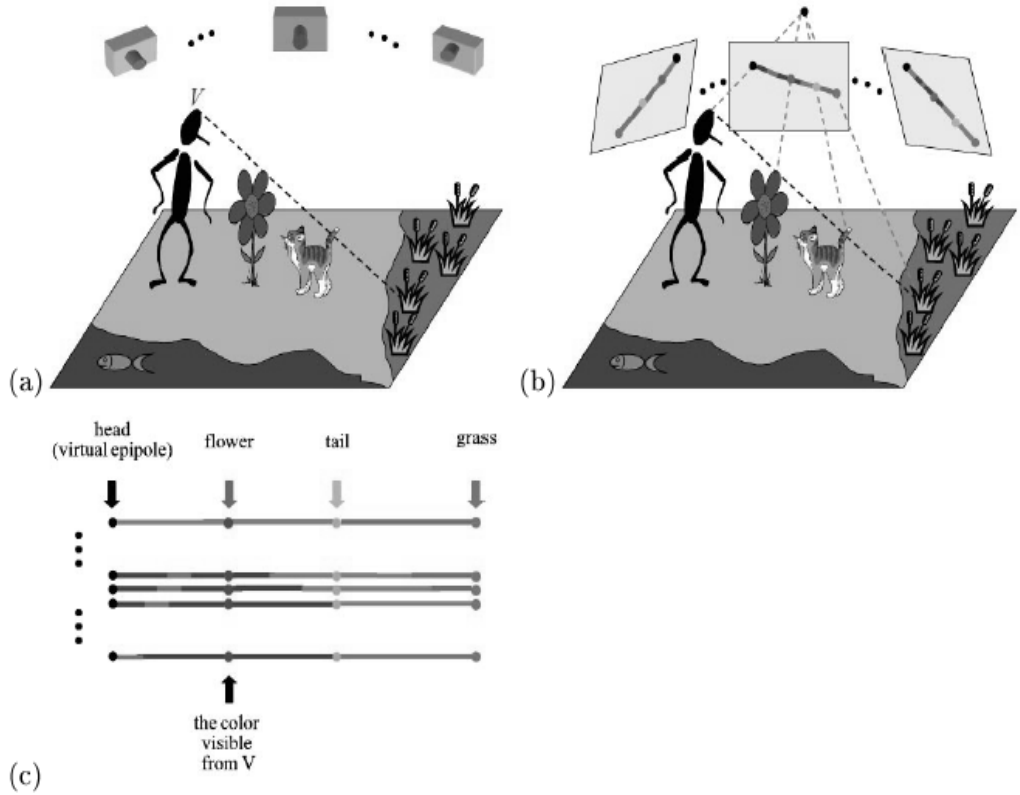

Figure 7.2: (a) The scene is shot using n cameras. The point of interest is V. (b) Projection of 3D LOS onto input cameras. (c) The projected lines are aligned and stacked. The color consistent points are flower, tail and grass. Since flower is closest to V, color of flower is selected as the color visible from point V.

An interesting work by Irani *et al.* proposes to reconstruct views at any point in the scene *without* computing any correspondence estimation [30]. In their algorithm, they first align and compare all the projections of each *line of sight* emerging from the virtual camera center in the input views and then the they create the virtual view. They describe the line-of-sight as the sight of the viewpoint along the line that stretches from the point to the direction desired.

To better understand this method, consider the scene given in Fig. 7.2 [30] where n uncalibrated cameras are utilized. The question is how scene looks like from the eye of the person, point V. In order to extract the color information which is visible from V, the line of sight (LOS) stretching from V should be analyzed. In this example, the first physical point that LOS intersects is the flower, therefore visible from V. This 3D LOS currently analyzed creates a 2D projection on each image as shown in Fig. 7.2.b. Therefore, the idea is to find the 2D projections of a 3D LOS and to

stack them as shown in Fig. 7.2.c to create color-consistent columns. Next step is to choose the closest column to viewpoint and assign as the color visible from point V in the direction of the LOS.

Although it is a bit different than view synthesis, we would like to mention the work of Seitz and Dyer [62]. Motivation of view morphing is in creating natural additional views between two images which are going to be used in transition from one image to the other when played as a video sequence. Therefore, images do not necessarily have to be the same. Due to its ability to change the viewpoint, view morphing is somehow related to view synthesis and can be used for this purpose.

Another approach in intermediate view reconstruction used especially in computer vision area is to reconstruct a 3D volumetric model of the scene and then to use this model to map new virtual views. All methods we have already mentioned were 2D methods. They first relate the images with correspondence estimation and then work towards view reconstruction.

Volumetric methods first create a 3D model of the scene and then manipulate and transform the object and easily create new views. However, these methods' execution speed is dependent on scene complexity. Moreover, they require sophisticated software and hardware for a realistic result.

There are also other works that use image sequences instead of calibrated multiple cameras [21].

Although they are a bit different from intermediate view reconstruction, it is going to be appropriate to mention other image-based rendering techniques. A survey by Kang [32] gives details about image-based rendering techniques. The term image-based is used here to describe that the methods use explicit images rather than volumetric 3D models.

The first category is non-physically based image mapping. One of the ideas in this method is to create an arbitrary view as a linear combination of a set of images which are considered to be basis images. This method is also utilized in entertainment industry for view morphing.

Another category is mosaicing. The aim in mosaicing is to create a higher resolution or a larger image using the combination of at least two different images. An example of this work is to create panoramic image using a few images. Since intermediate views are also contained in mosaics, arbitrary view generation is also possible. In order to create the image mosaics from images, the constituent images are registered first. Many image registration techniques are used for registering. One of the main obstacle in this method is to remove the discontinuities and distortions at boundaries where images are connected [32].

The third category of methods is interpolation from dense samples. The aim here is to first build up a lookup table by many samples from a scene from many cameras and then arbitrary view is generated by interpolating the stored lookup table. The advantage of this method is that correspondence estimation step is usually not necessary. Rapid image synthesis is possible using the lookup table without the estimation step. However, the number of samples required is very large (in hundreds

and thousands). In addition, camera viewpoints should also be available. These are the main drawbacks of this method [32].

Chapter 8

Conclusion and future work

In this report, we first presented how humans perceive the depth and described the basic monocular and binocular depth cues used by the brain. We then described the technologies used for stereoscopic viewing and mentioned the SynthaGram^{TM} monitor which utilizes lenticular sheets. SynthaGram monitors do not require the viewer to use special glasses and, therefore are more attractive to the audience. They can also deliver look-around feeling which is not possible in ordinary stereoscopic technologies.

The geometry of 3D was described next. We related the points in real world with points in an image. Following this projection we introduced stereo cameras and elaborated on the projection of the same 3D point onto image planes. The displacement of the point between its projections, which is called disparity, was defined mathematically.

Following this, we discussed the basic methods used for correspondence estimation. Block matching, although being very simple to implement, suffer from the unrealistic assumption that all the points in the same block should have the same disparity (or depth) value.

Feature matching methods which can yield good results, suffer from the feature extraction part. In case of unreliable features, results will be degraded. Another problem is that these methods create sparse correspondence maps.

Optical flow methods are the first methods that consider the image as a whole and try to estimate correspondence map as a function that can describe the motion. They give very promising results, however they can only describe displacements of a few pixels and therefore not suitable for most cases. Moreover, the discretization step of the continuous functions can significantly affect the results.

One of the most promising emerging methods is the use of partial differential equations to estimate the correspondences. They yield very good results when compared to other methods.

Next, we described the methods that create new virtual views. We mainly concentrated on 2D methods, which explicitly work on images. They first relate the images using correspondence estimation methods and then use this result for view synthesis. There are also 3D methods which first reconstruct a 3D model of the scene and then map back to 2D to create a virtual view.

8.1 Future work

We plan to use more than two cameras to shoot the scene and then try to compute additional intermediate views. As we presented before, one of the main obstacles in correspondence estimation are occlusions. In case of occlusions, two cameras will not be enough to extract depth information. Introducing extra cameras into acquisition process may solve this problem. Following the detection of occlusions, we can ignore the information from problematic camera and use other cameras for reliable information.

Another advantage of having multiple cameras is that we will have additional constraints during the computation of depth and creating intermediate views. Consider the case when there is no occlusion. Giving the emphasis to the central camera, we can compute many different depth values for the same point using the central camera and one of the additional cameras. The very important result here is that all depth values should be exactly equal since the cameras are coplanar. We may use this constraint for better estimation and reconstruction.

Following the reconstruction of the depth map, we would like to fit a continuous function onto this discrete map. We can utilize one of the works [71, 70] on this topic. The advantage of fitting a continuous function is that in case of mapping to a new viewpoint, since we have a depth everywhere, no holes will exist in the new depth map.

8.2 Conclusion

In this report, we have described some of the methods found in the literature for correspondence estimation and intermediate view reconstruction. We mentioned pros and cons of each method. Finally, we set our future plan.

Bibliography

[1] "*Facts About Holograms.*" http://www.talion.com /hologram.htm.

[2] "*History of 3-D Photography and Anaglyphs.*" http://www.sue-davis.net /e23d/history.html.

[3] "*History of Stereo Imagery.*" http://www.threeguesses.com/learn/history.php.

[4] "*MacNaughton Inc..*" http://www.nuvision3d.com.

[5] "*Moir Interferometry.*" http://aemes.mae.ufl.edu/ pgi/projects.htm.

[6] "*Phenomena of Binocular Vision.*" http://www.stereoscopy.com/library/wheatstone-paper1838.html.

[7] "*Stereographics Inc..*" http://www.stereographics.com.

[8] "*Virtual Empire: What is Stereo Photography?.*" http://www.usyd.edu.au/ su/macleay/01VE/veb1/veb1origins.html.

[9] M. Abdel-Mottaleb, R. Chellappa, and A. Rosenfeld, "Binocular motion stereo using MAP estimation," in *In Proc. of Computer Vision Pattern Recognition*, pp. 321–327, 1993.

[10] M. Accame, F. G. B. D. Natale, and D. Giusto, "Hierarchical block matching for disparity estimation in stereo sequences," in *Proc. International Conf. on Image Process.*, pp. 2374–2377, 1995.

[11] U. Ahlvers, U. Zlzer, and S. Rechmeier, "FT-based disparity estimation for stereo image coding," in *Proc. of IEEE Int. Conf. on Image Processing*, Sept. 2003.

[12] L. Alvarez, R. Deriche, J. Snchez, and J.Weickert, "Dense disparity map estimation respecting image discontinuities : A PDE and scale-space based approach," *Journal of Visual Communication and Image Representation*, vol. 13, pp. 3–21, 2002.

[13] L. Alvarez, J. Weickert, and J. Sanchez, "Reliable estimation of dense optical flow fields with large displacements," *International Journal of Computer Vision*, vol. 39, pp. 41–56, 2000.

[14] J. Barron, D. J. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[15] A. Baumberg, "Reliable feature matching across widely separated views," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2000.

[16] J. Bouguet, "Camera calibration toolbox for Matlab," Apr. 2002.

[17] F. M. Candocia and M. Adjouadi, "A similarity measure for stereo feature matching," *IEEE Trans. Image Process.*, vol. 6, pp. 1460–1464, Oct. 1997.

[18] V. Caselles, J. Morel, G. Sapiro, and A. Tannenbaum, "Introduction to special issue of PDE and geometry-driven diffusion in image processing and analysis," *IEEE Trans. Image Process.*, vol. 7, pp. 269–273, Mar. 1998.

[19] T. Chan, J. Shen, and L. Vese, "Variational pde models in image processing," *Notice of Amer. Math. Soc.*, vol. 50:14-26, January 2003.

[20] M. Chang, M. Sezan, and A. Tekalp, "An algorithm for simultaneous motion estimation and scene segmentation," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, pp. V.221–V.224, Apr. 1994.

[21] 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.

[22] A. Cozzi, B. Crespi, F. Valentinotti, and F. Wrgtter, "Performance of phase-based algorithms for disparity estimation," *Machine Vision and Applications*, vol. 9, no. 5-6, pp. 334–340, 1997.

[23] L. Falkenhagen, "Block-based depth estimation from image triples with unrestricted camera setup," in *Proc. IEEE Multimedia Signal Processing Workshop*, pp. 23–25, June 1997.

[24] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint.* Cambridge, MA: MIT Press, 1993.

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

[26] P. Havaldar, M. Lee, and G. Medioni, "View synthesis from unregistered 2D images," in *Graphics Interface I96*, pp. 61–69, 1996.

[27] J. Hedgecoe, *The Photographer's Handbook.* Knopf, 1992.

[28] B. Horn, *Robot vision.* Cambridge, MA: MIT Press, 1986.

[29] B. Horn and B. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–203, 1981.

[30] M. Irani, T. Hassner, and P. Anandan, "What does the scene look like from a scene point?," in *Proc. European Conf. on Computer Vision*, vol. 2, pp. 883–897, 2002.

[31] C. W. J. Wiklund and H. Knutsson., "Hierarchical phase based disparity estimation," in *Proc. Second Int. Singapore Conf. on Image Process.*, pp. 128–131, Sept. 1992.

[32] S.-B. Kang, "A survey of image-based rendering techniques," Tech. Rep. CRL 97/4, Digital Equipment Corp., Cambridge Research Lab, Aug. 1997.

[33] J. Konrad, "View reconstruction for 3-D video entertainment: issues, algorithms and applications," in *Proc. Int. Conf. on Image Process. and its Applications*, pp. 8–12, July 1999.

[34] J. Konrad, "Visual communications of tomorrow: natural, efficient and flexible," *IEEE Comm. Mag.*, vol. 39, pp. 126–133, Jan. 2001.

[35] J. Konrad and P. Agniel, "Artifact reduction in lenticular multiscopic 3-D displays by means of anti-alias filtering," in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, vol. 5006A, pp. 336–347, Jan. 2003.

[36] J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 910–927, Sept. 1992.

[37] J. Konrad and Z.-D. Lan, "Dense disparity estimation from feature correspondences," in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, vol. 3957, pp. 90–101, Jan. 2000.

[38] L. Lipton, *StereoGraphics Developer's Handbook*. StereoGraphics Corporation, 1991.

[39] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 2, pp. 148–157, 1993.

[40] H. Longuet-Higgins, "The reconstruction of a scene from two projections – configurations that defeat the 8-point algorithm," in *Conference on Artificial Intelligence Applications*, pp. 395–397, 1984.

[41] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artificial Intell.*, pp. 674–679, Aug. 1981.

[42] A. Maki, L. Bretzner, and J.-O. Eklundh, "Local Fourier phase and disparity estimates: An analytical study," *Lecture Notes in Computer Science*, vol. 970, pp. 868–873, 1995.

[43] A. Maki, T. Uhlin, and J. Eklundh, "Phase based disparity estimation in binocular tracking," in *Proc. 8th Scandinavian Conference on Image Analysis*, 1993.

[44] A. Mancini, "Disparity estimation and intermediate view reconstruction for novel applications in stereoscopic video," Master's thesis, McGill University, Dept. Electr. Eng., Feb. 1998.

[45] A. Mancini and J. Konrad, "Robust quadtree-based disparity estimation for the reconstruction of intermediate stereoscopic images," in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, vol. 3295, pp. 53–64, Jan. 1998.

[46] A.-R. Mansouri and J. Konrad, "Bayesian winner-take-all reconstruction of intermediate views from stereoscopic images," *IEEE Trans. Image Process.*, vol. 9, pp. 1710–1722, Oct. 2000.

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

[48] B. McCane, K. Novins, D. Crannitch, and B. Galvin, "On benchmarking optical flow," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 126–143, 2001.

[49] D. Mount, N. Netanyahu, and J. LeMoigne, "Efficient algorithms for robust feature matching," *Pattern Recognit.*, vol. 32, pp. 17–38, January 1999.

[50] H.-H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 565–593, Sept. 1986.

[51] 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.

[52] J.-I. Park, N. Yagi, and K. Enami, "Image synthesis based on estimation of camera parameters from image sequence," *IEICE Trans. on Information and Systems*, vol. E77-D, no. 9, pp. 973–986, 1994.

[53] J. Park and S. Inoue, "Arbitrary view generation using multiple cameras," in *Proc. International Conference on Image Process.*, vol. 1, pp. 149–153, Oct. 1997.

[54] J. Park and S. Inoue, "Acquisition of sharp depth map from multiple cameras," *Signal Process., Image Commun.*, vol. 14, pp. 7–19, November 1998.

[55] S. Pastoor and M. Wöpking, "3-D displays: A review of current technologies," *Displays*, vol. 17, pp. 100–110, 1997.

[56] M. Proesmans, L. V. Gool, E. Pauwels, and A. Oosterlinck, "Determination of optical flow and its discontinuities using non-linear diffusion," in *3rd Eurpoean Conference on Computer Vision*, vol. 2, pp. 295–304, 1994.

[57] A. Redert, E. Hendriks, and J. Biemond, "Synthesis of multi viewpoint images at non-intermediate positions," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. IV, pp. 2749–2752, Apr. 1997.

[58] A. Redert, E. Hendriks, and J. Biemond, "Correspondence estimation in image pairs," *IEEE Signal Process. Mag.*, vol. 16, no. 3, pp. 29–46, 1999.

[59] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.

[60] D. Scharstein, "Matching images by comparing their gradient fields," in *Proc. of 12th International Conference on Pattern Recognition*, Oct. 1994.

[61] D. Scharstein, "Stereo vision for view synthesis," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, June 1996.

[62] S. Seitz and C. Dyer, "View morphing," in *Proc. SIGGRAPH 96*, pp. 21–30, 1996.

[63] A. Sethuraman, M. Siegel, and A. Jordan, "A multiresolution framweork for stereoscopic image sequence processing," in *Proc. IEEE Int. Conf. Image Processing*, pp. 361–365, Nov. 1994.

[64] S.Sharghi and F. A. Kamangar, "Geometric feature-based matching in stereo images," in *Proceedings of Information Decision and Control*, pp. 65–70, Feb. 1999.

[65] R. Stasiński and J. Konrad, "POCS-based image reconstruction from irregularly-spaced samples," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 315–318, Sept. 2000.

[66] C. Strecha and L. V. Gool, "Pde-based multi-view depth estimation," in *1st International Symposium of 3D Data Processing Visualization and Transmission*, vol. 2, pp. 416–425, 2002.

[67] C. Van Berkel, A. Franklin, and J. Mansell, "Design and applications of multi-view 3D-LCD," in *Proc. SID Euro-Display '96*, pp. 109–112, 1996.

[68] A. Woods, T. Docherty, and R. Koch, "Image distortions in stereoscopic video systems," in *Proc. SPIE Stereoscopic Displays and Applications*, vol. 1915, Feb. 1993.

[69] L. Zhang, "Hierarchical block-based disparity estimation using mean absolute difference and dynamic programming," in *Proc. International Workshop on Very Low Bitrate Video Coding*, 2001.

[70] H.-K. Zhao, , S.Osher, and R. Fedkiw, "Fast surface reconstruction using the level set method," in *1st IEEE Workshop on Variational and Level Set Methods*, 2001.

[71] H.-K. Zhao, S.Osher, B. Merriman, and M. Kang, "Implicit and non-parametric shape reconstruction from unorganized data using a variational level set method," *Computer Vision and Image Understanding*, vol. 80, pp. 295–314, 2000.

[72] M. Ziegler, *Region-based analysis and coding of stereoscopic video.* PhD thesis, Thesis Technische Universiteit Delft, 1997.