2001

Dense matching of uncalibrated images for stereo vision.

Hongxuan. Jin
University of Windsor

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Dense Matching of Uncalibrated Images for Stereo Vision

HONGXUAN JIN

A Thesis
Submitted to the Faculty of Graduate Studies and Research
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2001

Abstract

Stereo vision is aimed at recovering 3D structure from two images taken with cameras positioned at different viewpoints. To obtain the depth of a scene, we need to establish the correspondence of pixels or features between the two stereo images. This process is called matching. Dense matching uncalibrated images is a difficult task; it requires the matching of each and every pixel between images, while no knowledge of the camera parameters is available.

Most existing methods for dense matching uncalibrated images are impractical and time consuming. In order to develop a fast, accurate and practical method, we attempted to use image interest points to provide a disparity estimate. Then we proposed a fast dense matching algorithm which integrates edge features of the image. The matching was carried out separately for edge areas and non-edge areas. In order to match the non-edge areas of the image, matching constraints were combined to restrict the search region. This approach effectively reduces the computational time and improves the matching quality. Our hybrid method has been tested on several indoor and outdoor scenes and the results demonstrate its capability.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Computer Vision

Computer vision was born of the desire to create intelligent machines. In the 1950's, Alan Turing asked the question 'Can a machine think?', which led to the development of artificial intelligence(AI). The goal of AI is to simulate natural biological systems. An intelligent machine, or a robot system, should be able to work autonomously within its environment. The visual capability of a machine has long been regarded as the one with the highest potential, because most animals use vision as their most prominent way of gathering information. To give machines this ability of perceiving visual cues is the aim of computer vision. In other words, the goal of computer vision is to let a computer see.

For many animals and humans, the process of obtaining visual information is based on pictures sensed on the retina. Similarly, a computer(machine) gathers information based on digital images taken by visual sensors, typically cameras. In [29], the scope of computer vision is defined as: *a set of computational techniques aimed at estimating or making explicit the geometric and dynamic properties of the 3D world from digital images.* In [27], another definition is given as the following: *The goal of computer vision is to make useful decisions about real physical objects and scenes based on sensed images.*

The field of computer vision is vast and in continuous expansion. Object detection, motion analysis, pattern recognition and three-dimensional(3D) reconstruction are some of its research areas. One major topic in computer vision discipline is stereo vision.

1.2 Stereo Vision

The recovery of 3D structure of a scene from 2D images is one important goal in visual processing. Different sources can be used; for example, the shading in an image reveals information about 3D shapes. However, shape from shading is not reliable and is greatly influenced by lighting conditions. Another more important and reliable source of 3D information is provided by the change in location of an object from one image to another, which is called *stereo vision*, or *stereopsis*.

A stereo vision system aims at obtaining information about the 3D structure of the scene from stereo images. In the same way humans use a pair of eyes and brain to perceive the 3D world, a typical stereo vision system consists of a pair of cameras and a central processor. When we observe an object in a 3D scene, this object produces two images at different retinal locations of two eyes. Our brain is capable of measuring the difference and uses it to estimate the depth of the object. Similarly, a stereo vision system shall be able to obtain the 3D information from two images taken from different viewpoints; these two images are referred to as *a stereo image pair*.

Why do we need a pair of stereo images? As we know, humans need to use both eyes to get depth information from the 3D world around us. A simple experiment will illustrate this: when one eye is covered, the scene we observe becomes 'flat'. For the same reason, at least two images are needed to recover the 3D structure using a stereo vision system. As shown in figure 1.1, suppose that we try to find the depth of the point P in space that projects on the image point p in the left image. Since any point along the ray formed by Op could have projected at p, it is impossible to recover the depth of P based on only this image. However, with the help of another image(the right image), the depth of P can be easily calculated given its projection $p\prime$ and p in two images.

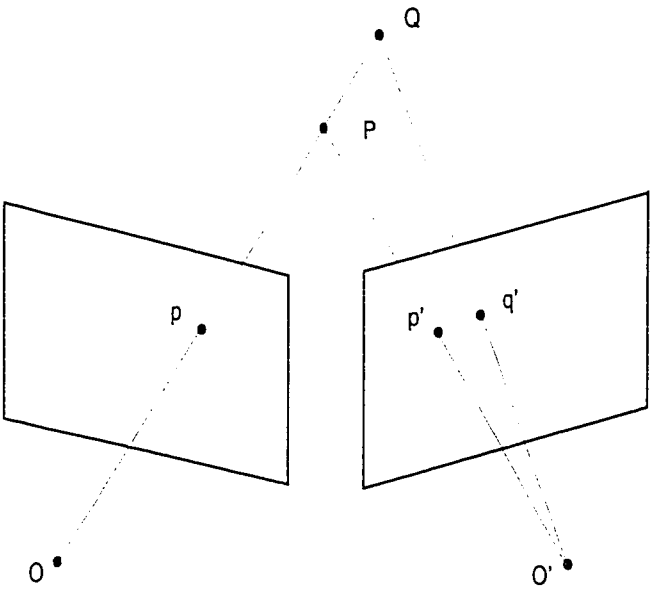


Figure 1.1: Why at least two images are required in stereo vision.

1.3 Problem of Matching

Several steps are involved in obtaining the 3D(or depth) information of a scene for a stereo vision system. To find out the depth of a particular point in one of the images, the following steps might be carried out:

- **Step 1**: Calibrate the stereo vision system. In other words, calibrate both cameras so that the geometrical relationships between the 3D space and the two cameras become known(calibration process);
- **Step 2**: Identify the image point that represents the same scene point in the other image(matching process);
- **Step 3**: Calculate the depth of the selected location based on the location difference of the corresponding points and camera positions(reconstruction process).

Without the calibration process, the above steps represent two problems: *correspondence* and *reconstruction*. The problem of correspondence is also known as the

matching problem and might involve more than two images. The main task of matching is to determine which part of the two images are projections of the same scene element. The location difference of two corresponding points from two images is called *disparity*. The reconstruction process involves obtaining the 3D location and structure of observed objects, given their matched observations in two or more images.

Determining the corresponding pixels and other features between stereo images represents a fundamental problem in stereo vision. It is a very difficult task, and it is the basis of the reconstruction process. Indeed, once matching is achieved, calculation of depth becomes a straightforward geometrical problem. A large amount of work has been carried out over the past two decades on the matching problem. The methods proposed can roughly be grouped into two categories:

- **Area-based methods**
  They are also known as template matching methods. In this category, the correspondences between images are determined based on the similarity of the pixel's gray value using image template windows.

- **Feature-based methods**
  In this category, correspondences are established between some selected features extracted from the images, such as edges, lines, or curves.

1.4 Dense Matching of Uncalibrated Images

To match a small number of pixels between the stereo pair is called *sparse matching*. In contrast, *dense matching* involves matching all pixels in the images. Dense matching is essential to many applications such as 3D reconstruction and view synthesis.

Images are *uncalibrated*; namely, the motion between them and the camera parameters are not known. In contrast to uncalibrated images, calibrated images are images taken by cameras with known position, which can facilitate the matching process.

Dense matching is one of the bottlenecks in stereo vision, since it involves a large amount of computations. The existence of occlusions makes the problem even more

difficult. To dense match uncalibrated stereo images, where the only information available is the raw images themselves, we need to find ways to assign each pixel from one image to its corresponding pixel in the other image. Figure 1.2 shows an uncalibrated stereo image pair.

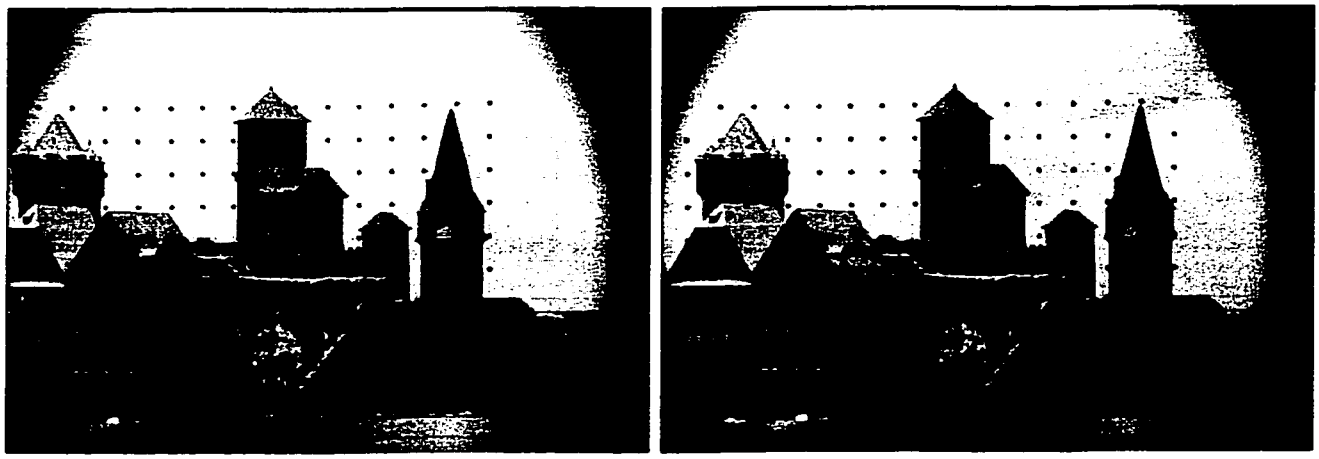


Figure 1.2: Stereo image pair: Castle(size 576 × 384).

This challenging problem of dense matching uncalibrated images is addressed in this thesis. The research objectives have been identified as the following:

- Analysis of the existing methods for achieving a dense match between uncalibrated stereo image pairs
- Experimental study of a simple exhaustive approach and improved algorithm using interest points
- Design and implementation of a fast dense matching algorithm which integrates image edge features
- Experimental and theoretical evaluation of the proposed algorithm

1.5 Thesis Overview

This thesis is organized as follows. Chapter 2 provides the theoretical background of issues related to stereo matching. Chapter 3 presents a review of previous approaches for obtaining dense matching from uncalibrated images; their advantages and disadvantages are outlined. Chapter 4 presents an experimental study of a simple exhaustive search approach and our improved algorithm using image interest points. Chapter 5 is the centerpiece of this thesis, where a hybrid algorithm aimed at exploiting edge feature is proposed; the experimental results of this hybrid algorithm demonstrate the improvement in accuracy and efficiency. Chapter 6 is a conclusion and discussion of directions for future research.

Chapter 2

Background for Stereo Matching

2.1 Digital Image and Camera Geometry

2.1.1 Digital Image

Like natural visual systems, image processing in computer vision starts with the reflection of light rays. When a light ray reflects off a surface and enters the camera, it hits a screen of sensor devices that registers the light intensity. In a CCD camera, the screen is an $n \times m$ array of CCD photo-sensors. From these sensors, an output of $n \times m$ different voltages is generated; then through a frame grabber, these voltage signals are digitalized and stored into an integer array of size $n \times m$. An $n \times m$ digital image is thus represented by this 2D integer array, with each entry of the array denoting the intensity(color) value of a pixel. The intensity value is usually represented by one byte for the gray-level images, and thus ranges from 0 to 255 (typically 0 is black, 255 white).

2.1.2 Camera Model

A real CCD camera presents many imperfections due to the optical distortion and digitalization process. Several geometric models have been devised to represent CCD cameras. These models differ in their complexities and in their closeness to the phys-

ical camera. The *pinhole model*, also known as the *full perspective model*, is a good approximation for a real camera and is by far the most used for modeling cameras.

When a light ray is reflected off an object surface and passes through the pinhole lens of a camera onto the sensor array, an image is obtained. This image is a *projection* of the object surface, it is also called the *image plane*. A point on the image plane is a pure perspective projection of a point in the scene, while an object in the scene is represented by a collection of pixels in the image.

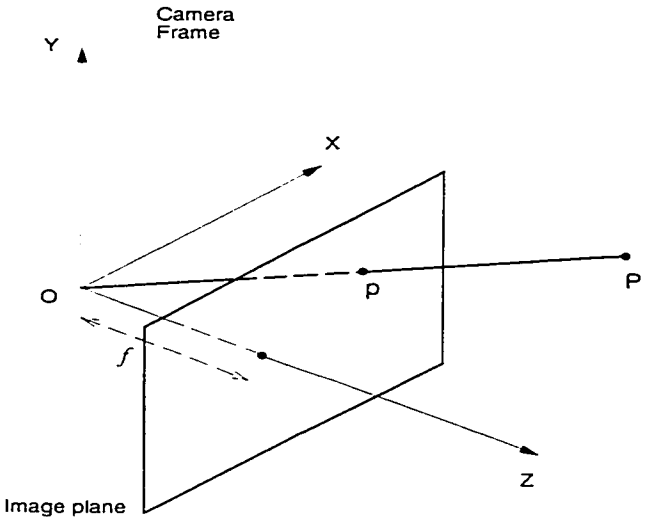


Figure 2.1: Camera geometry under the pinhole model.

Figure 2.1 shows the pinhole camera model. O is the *projection center*, which is drawn behind the image plane to make it easy to understand the projection model; the line through O and perpendicular to the image plane is the *optical axis* Z; the distance between image plane and projection center is the *focal length* f. The image of the scene point P is the image point p at which the straight line through P and O intersects the image plane.

Consider the 3D reference frame in which O is the origin and the image plane is orthogonal to the Z axis; let $P = [X, Y, Z]^\top$ and $p = [x, y, z]^\top$, this reference frame is called the *camera reference frame*. In this case we can write:

$$\begin{aligned} x &= f\frac{X}{Z}, \\ y &= f\frac{Y}{Z} \end{aligned} \tag{2.1}$$

2.1.3 Camera Parameters

The camera reference frame is used to write the fundamental equations (2.1) of the perspective projection . This camera reference frame is located within another reference frame called the *world reference frame*, which is also known as a *scene reference frame*. Directly from the images, we have the image coordinate system which is also called the *pixel reference frame*. The relationship between the camera, the image and the world coordinate system is shown in Figure 2.2.

The parameters linking the world coordinate system to image pixels are grouped into two categories. The *extrinsic parameters* of the camera define the location and orientation of the camera reference frame with respect to a world reference frame. The *intrinsic parameters* of the camera link the pixel coordinates of an image point to the corresponding coordinates in the camera reference frame.

Extrinsic Parameters

Finding the location and orientation of the camera with respect to the world reference frame is a common problem. Since the camera reference frame is often unknown, the extrinsic parameters can be defined as a set of geometric parameters that identify the transformation between the unknown camera reference frame and the world reference frame. This transformation is described by a 3D translation vector T and a 3D rotation matrix R. T describes the relative position of the origin of the world reference system within the camera reference frame. R, a rotation, is a 3×3 orthogonal matrix that brings the corresponding axes of the two frames onto each other. Given a point P in the scene which is referred to as P_{world} in the world frame and P_{camera} in the camera frame, the relationship between them can be defined in equation (2.2).

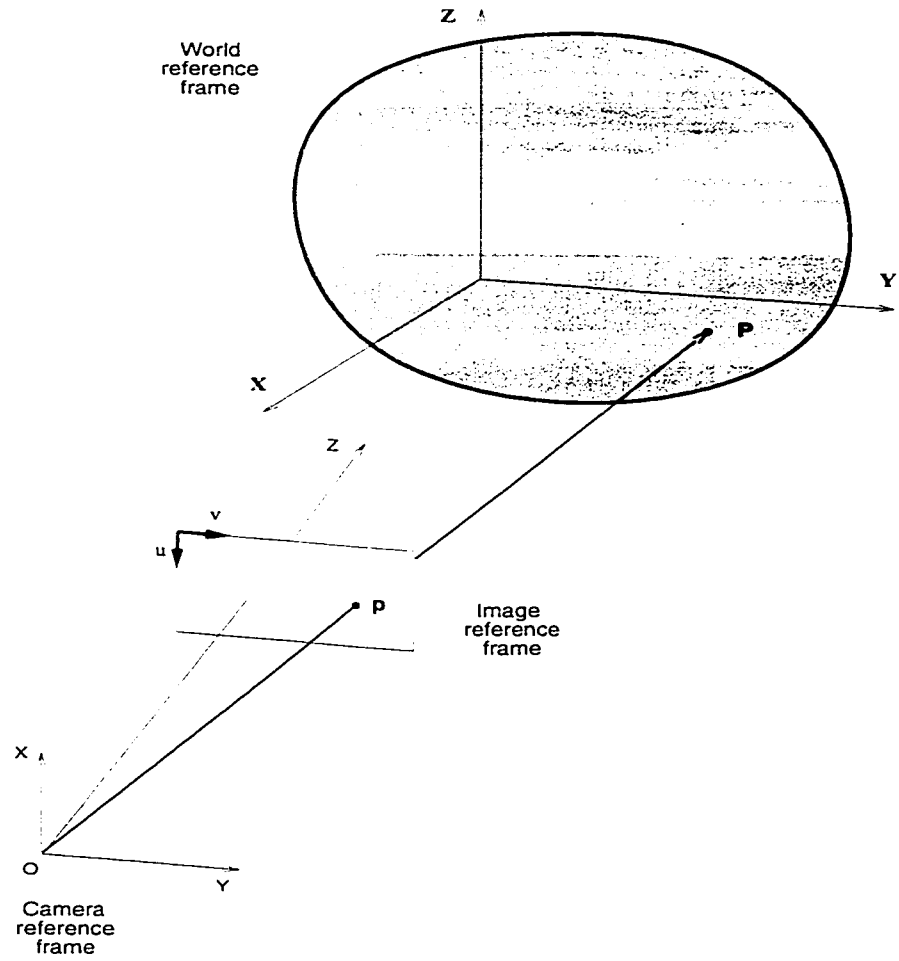


Figure 2.2: The relationship between camera, image and world coordinate system.

$$P_{camera} = R(P_{world} - T) \tag{2.2}$$

where P_{camera} and P_{world} are the 3D coordinates of a space point P given in the camera reference frame and in the world reference frame respectively.

Intrinsic Parameters

The intrinsic parameters are parameters required to characterize the optical, geometric and digital characteristics of a camera. For a pinhole camera, intrinsic parameters include: the focal length f, the transformation between camera coordinates and pixel coordinates, and also the geometric distortion introduced by optics.

Ignoring distortion, we have the equation (2.3) to link the coordinates of an image point denoted by (x_{img}, y_{img}) in pixel units, with the same point with coordinates (x, y) in the camera reference frame units.

$$\begin{aligned} x &= -(x_{img} - o_x)s_x, \\ y &= -(y_{img} - o_y)s_y \end{aligned} \tag{2.3}$$

where o_x, o_y are the coordinates of the image center, and (s_x, s_y) the pixel size in the horizontal and vertical direction respectively. Therefore, the set of intrinsic parameters is f, o_x, o_y, s_x, s_y.

Note that the minus sign is there because the projection center is actually in front of the image plane, which causes the world coordinates to change sign when projected on the image plane.

2.1.4 Camera Calibration and Uncalibrated Image

The perspective projection that transforms space points defined in the world coordinate system, into image points defined in the image coordinate system, can be described by

a 3×4 matrix, often denoted by M. The expression of this matrix in terms of intrinsic parameters, extrinsic parameters, and pure perspective projection is given by

$$M = AID \tag{2.4}$$

where

- A is the matrix for the intrinsic parameters given by

$$A = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.5}$$

- I is the matrix for the pure perspective projection given by

$$I = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.6}$$

- D is the matrix for the extrinsic parameters given by

$$D = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \tag{2.7}$$

The 3×3 matrix A depends only on the intrinsic parameters; it performs the transformation between the camera and the image reference frames. The 3×4 matrix I represents the perspective projection in the same reference frame. The 3×4 matrix D represents displacement between the world and the camera reference frames.

The projection of the space point $P = (X, Y, Z, 1)$ on the image point $p = (x, y, 1)$ is given by:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = AID \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.8}$$

where λ is a scale factor.

The problem of *camera calibration* is in estimating the values of the intrinsic and extrinsic parameters of the camera model. In other words, the calibration process means calculating the 3×4 matrix M. Since this process requires the use of a known calibration pattern, it is often difficult to calibrate a camera in practice. New methods have been proposed to overcome this problem. In particular, camera self-calibration is a new promising method being currently investigated by numerous researchers. The remainder of this thesis will consider only *uncalibrated images*; that is, images taken with cameras on which no calibration has been performed.

2.2 Stereo Matching

2.2.1 Definition

Image correspondence can be defined as a mapping between two images, both spatially and with respect to intensity. If the two images are denoted by I_1 and I_2, where $I_1(x, y)$ and $I_2(x', y')$ are image coordinates which map to the intensity value of the according pixel, then the matching between these two images can be expressed as:

$$I_2(x', y') = g\left(I_1(f(x, y))\right) \tag{2.9}$$

where g is a 2D intensity transformation, and f is a 2D spatial-coordinate transformation. g should be considered when two images are taken with different types of

sensors; it is not necessary in a typical stereo vision system with only CCD cameras. Therefore, matching is to find a transformation f that maps spatial coordinates x and y, to new spatial coordinates $x\prime$ and $y\prime$ as shown in equation (2.10).

$$(x\prime, y\prime) = f(x, y) \tag{2.10}$$

Some typical geometric transformation between the two images are shown in Figure 2.3. Translation transformation occurs when the images are misaligned by a small shift due to a change in the camera's position. Rotation transformation is caused by a camera rotation around the axis. Rigid transformations are those where the objects in the images retain their relative size and shape.

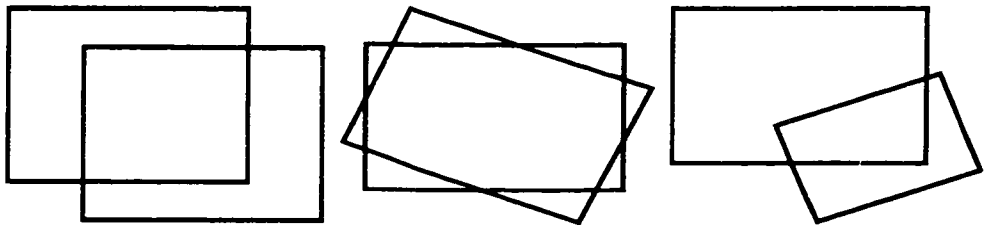


Figure 2.3: Typical geometry transformation: (a) translation. (b) rotation. (c) rigid transformation.

If the transformation between two images is 2D, the spatial transformation f can be expressed by a single equation that maps each point in the first image to a new location in the second image. However, because the 3D-2D perspective projection is an irreversible one, it is impossible to find a spatial mapping function between a pair of stereo images. Therefore, stereo matching is defined as locating a pair of image points resulting from the projection of the same object point by some similarity constraints of the pixel colors. The relation can be written as follows:

$$I_2(x', y') = I_1(x + d_x, y + d_y) \tag{2.11}$$

where d_x and d_y are the location differences of the matching pairs along the x and y coordinates respectively.

Note that not all the points in one image can find their corresponding points in the other image; this is called *occlusion*. It is because a given object point may not have a projection in both images. For instance, a point may appear in the left image but not in the right image because it becomes hidden due to the position and orientation of the right camera.

2.2.2 Disparity and Disparity Map

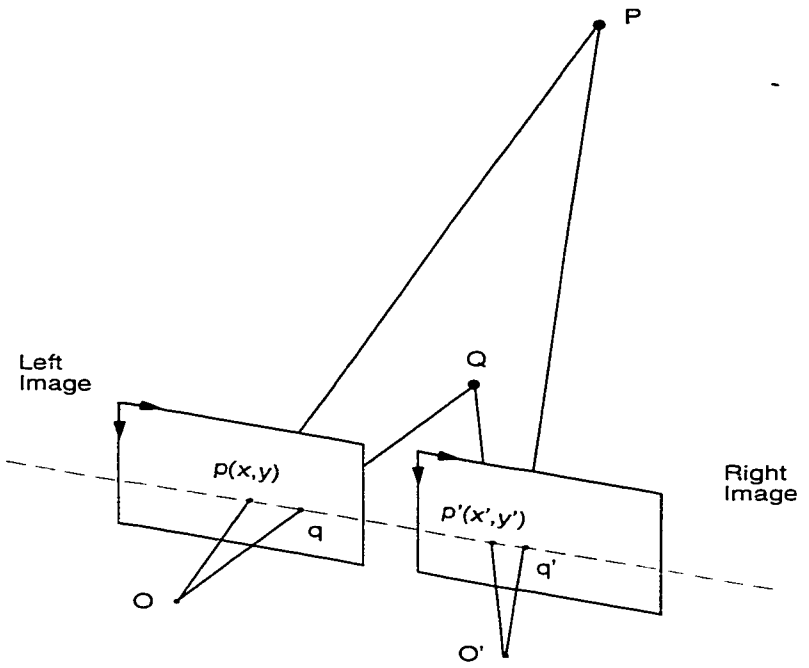


Figure 2.4: Disparity basics.

Disparity specifies the offset of a pixel in the first image to its match in the second image. Without loss of generality, we can assume that the two image planes are parallel. As shown in Figure 2.4, an object point P projects in the left image on $p(x, y)$ and in right image on $p\prime(x\prime, y\prime)$. Since the two image planes are parallel, we have $y = y\prime$. Therefore the disparity d of P on this pair of images is one dimensional, and $d = x\prime - x$.

Matching results can be stored in a *disparity map*, while the result of dense matching is recorded in a dense disparity map. A disparity map is formed by the disparities of all the matched points and can be displayed as an image. It is defined as an integer valued array where each entry stores the disparity value of the same image location. For example, if the disparity map D records the matching result between left and right images, given a pair of matched points $(p(x,y), p\prime(x\prime,y\prime))$, then $D(x,y) = k$ with $k = x\prime - x$. In that case, k is considered as the intensity value (gray value) of the pixel at (x,y) of the disparity map picture.

Figure 2.5 shows a pair of synthetic images, and their matching result is shown as a disparity map in Figure 2.6. This disparity map is also a *ground truth* disparity map, since the exact match result is known for synthetic images[1].

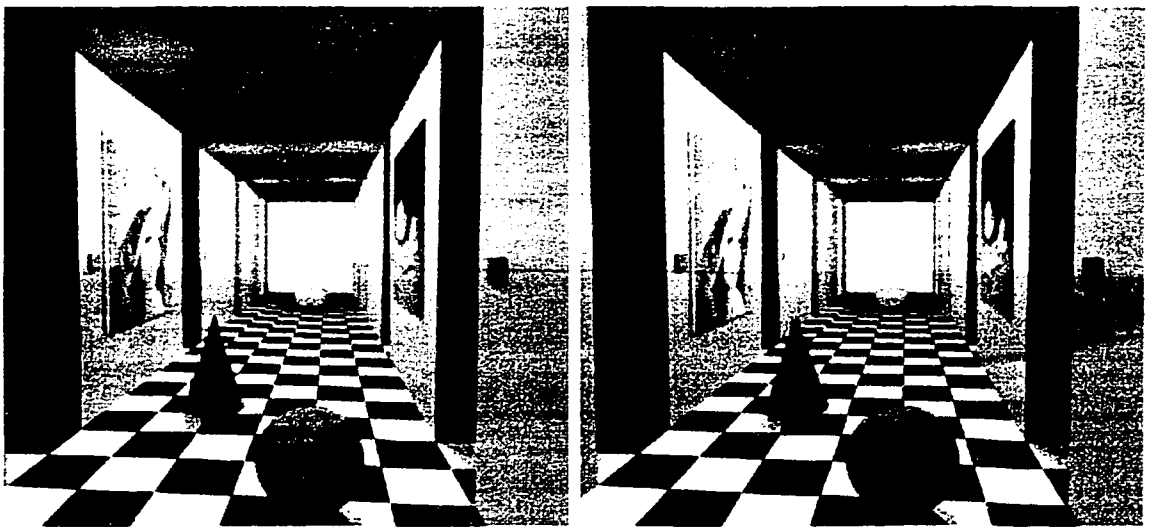


Figure 2.5: Example of a synthetic stereo image pair.

The disparity value indicates the distance from the cameras to the object point. If an object point is infinitely far away, then its projection onto the two image planes will be at the same location, and the disparity will be zero. If an object is close to the cameras then the disparity will be large. Disparity is inversely proportional to the distance between an object and the camera system. As shown in Figure 2.4, the

[1]Histogram equalization has been performed on the disparity map to increase image contrast.

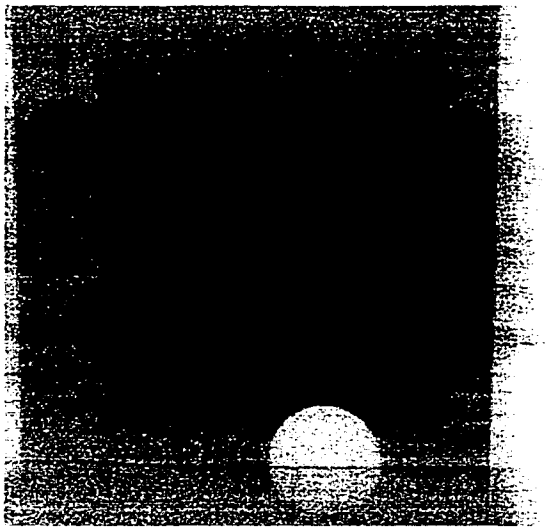


Figure 2.6: The ground truth disparity map.

disparity of object point P is less than the disparity of Q, since the location of Q is closer than P to the cameras. In the disparity map of Figure 2.6, the area is darker where it is deeper in depth along the corridor, which indicates smaller disparity values[2].

[2]Intensity value 0 is black, 255 white.

2.3 Stereo Matching Constraints

2.3.1 Epipolar Constraint

Epipolar Geometry

Epipolar geometry describes the geometrical relationship between a pair of stereo images. This relationship is shown in Figure 2.7. There are two pinhole cameras, with their projection centers O and $O\prime$ respectively, and two image planes. An object point P projects as p on the left and $p\prime$ on the right image. The projection of O on the right image and $O\prime$ on the left image are denoted respectively by e and $e\prime$, where e and $e\prime$ are called *epipoles*.

An *epipolar line* is defined by an image point and the epipole on the image plane. The line defined by e and p is an epipolar line, and its corresponding epipolar line is defined by $e\prime$ and $p\prime$. The plane defined by P, O and $O\prime$ is the *epipolar plane*. Epipolar lines can also be viewed as the intersection between the epipolar plane and the image planes.

Epipolar Constraint

Given a point p in the left image, the object point P that was projected on p may lie anywhere on the ray defined by O and p (recall that it is impossible to recover the depth of a point from a single image). However, the image of this ray in the right image is the epipolar line defined by the corresponding point $p\prime$ and the epipole $e\prime$. Therefore, the correct match of p must lie on this corresponding epipolar line in the right image. This constraint is known as the *epipolar constraint*; it establishes a mapping between points in the left image and lines in the right image and vice versa. If we assume that the epipolar geometry is known, the search for the match of p in the right image can be restricted to the search along the epipolar line of p. The matching problem is reduced from a two-dimensional search to a one-dimensional search. This is a considerable simplification of the problem; for instance, the search over $1000 \times 1000 = 10^6$pixels (a 1000×1000 image) will be reduced to a search over 1500 pixels(a diagonal line at

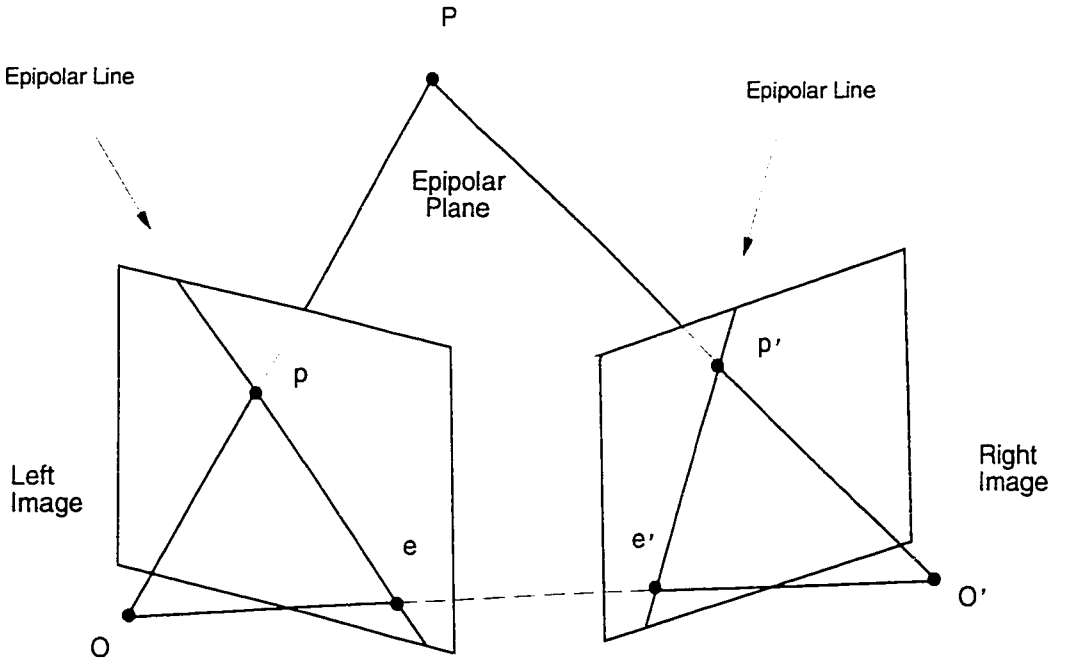


Figure 2.7: Epipolar geometry.

most).

Essential Matrix and Fundamental Matrix

The mapping between points in one image and epipolar lines in the other can be established by estimating two important matrices: the essential matrix E and fundamental matrix F.

In camera coordinates(*not* in pixel coordinates), for each pair of corresponding points u and $u\prime$, we have the 3×3 essential matrix E that satisfies the following equation:

$$u\prime^{\top} E u = 0 \tag{2.12}$$

This equation can be generalized to the pixel coordinate system when rewritten as:

$$u\prime^{\top} A^{\top} (A^{-1})^{\top} E A^{-1} A u = 0 \tag{2.13}$$

where A is the 3×3 intrinsic parameter matrix defined in a previous paragraph.

The above relation is equivalent to

$$(Au\prime)^{\top} (A^{-1})^{\top} E A^{-1} (Au) = 0 \tag{2.14}$$

Because Au is the same point expressed in the pixel coordinate system, then we have $Au = p$ and $Au' = p'$. The above relation becomes

$$p'^{\top} (A^{-1})^{\top} E A^{-1} p = 0 \tag{2.15}$$

If we denote the 3×3 matrix $(A^{-1})^{\top} E A^{-1}$ by F, then we have

$$p\prime^{\top} F p = 0 \tag{2.16}$$

where F is called the *fundamental matrix*. $Fp = (a, b, c)$ represents the coefficients of the corresponding epipolar line in the right image, on which $p\prime$ should be located.

The significance of the fundamental matrix F is that equation (2.16) is defined in terms of pixels. If F is known, the matching process can be simplified from a 2D to a 1D problem as stated previously. In equation (2.16), F can be computed when given 8 or more matched points in a pair of uncalibrated images. This simple linear algorithm is know as the eight-point algorithm for calculating F. However, This eight-point algorithm is not stable and very sensitive to noise. More reliable and stable methods exist for the calculation of F; see for instance [15][5].

2.3.2 Other Constraints

Order Constraint

The order constraint states that stereo projections always preserve the order of points along the according epipolar line. As shown in Figure 2.8, if point n is on the right side of point m on the epipolar line, then the matching point $n\prime$ of n must lie on the right side of the matching $m\prime$ of m. The reason is that it is geometrically impossible for points projected from the same opaque surface to be differently ordered in the stereo image pair.

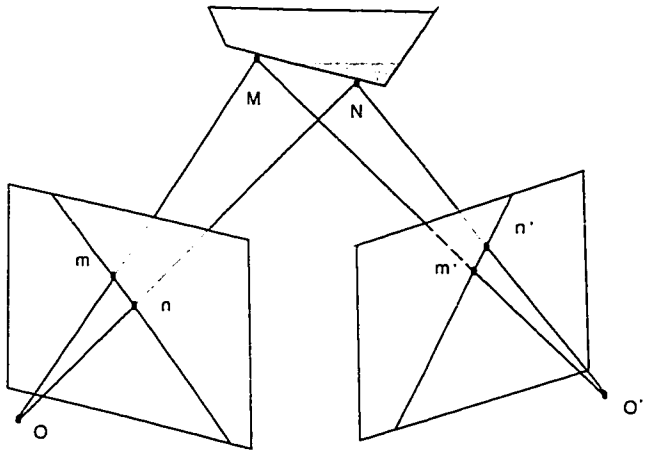


Figure 2.8: Order constraint.

Continuity Constraint

The continuity constraint is also known as surface smoothness constraint. The underlying idea of this constraint is that the world is mostly made up of objects with smooth surfaces. It states that disparity varies smoothly on object surfaces; sharp changes of disparity occur at object boundaries. In the ground truth disparity map(Figure 2.6), we can recognize the shape of the objects in the scene since the sharp intensity change at the object's boundary. This abrupt change indicates the disparity discontinuity at the object boundary. Meanwhile on the surface of the objects intensity changes smoothly and uniformly, which indicates the continuity of disparity values.

Uniqueness Constraint

The uniqueness constraint states that one image point has at most one match in the other image. It is impossible that one object point can project at more than one location in only one image; while there is no match in the case of occlusion. The uniqueness constraint can simplify the computation and can be used to validate the matching results.

2.4 Methods for Stereo Matching

Stereo matching methods can be viewed as a different combination of choices for the following three components:

- A matching token
- A similarity measurement
- A search strategy

A *matching token* represents the information in the images that will be used for matching. A basic matching token is the pixel's intensity value; while other tokens can be image features, such as edges and contours. A *similarity measurement* determines the measure of similarity for each test; different measurements apply to different types of matching tokens. A *search strategy* is how the search area is determined in the target image and how the search is actually carried out.

There are two major categories of stereo matching methods based on different matching tokens. The area-based methods and feature-based methods. In the area-based method, the matching element is template windows of a certain size. The feature-based method aims at establishing the correspondence between a set of image features based on some global optimization. Correlation functions are used to measure the similarity in area-based methods, while for feature-based methods more complicated criteria are adopted. Search strategies also vary for these two classes of methods, although some principles apply to both.

2.4.1 Area-based Method

In area-based methods, the matching is carried out by calculating and comparing correlations between template windows. The correlation process is the essential part for an area-based matching algorithm. To match a pixel p from the left image in the right image, a small window(reference window) is located with p as the center. This window is then compared with same sized windows(target window) in the right image for each pixel in the search area. Each comparison produces a correlation score using certain correlation functions. The corresponding pixel $p\prime$ shall be associated with the window that maximizes the similarity function. This process is shown in Figure 2.9.

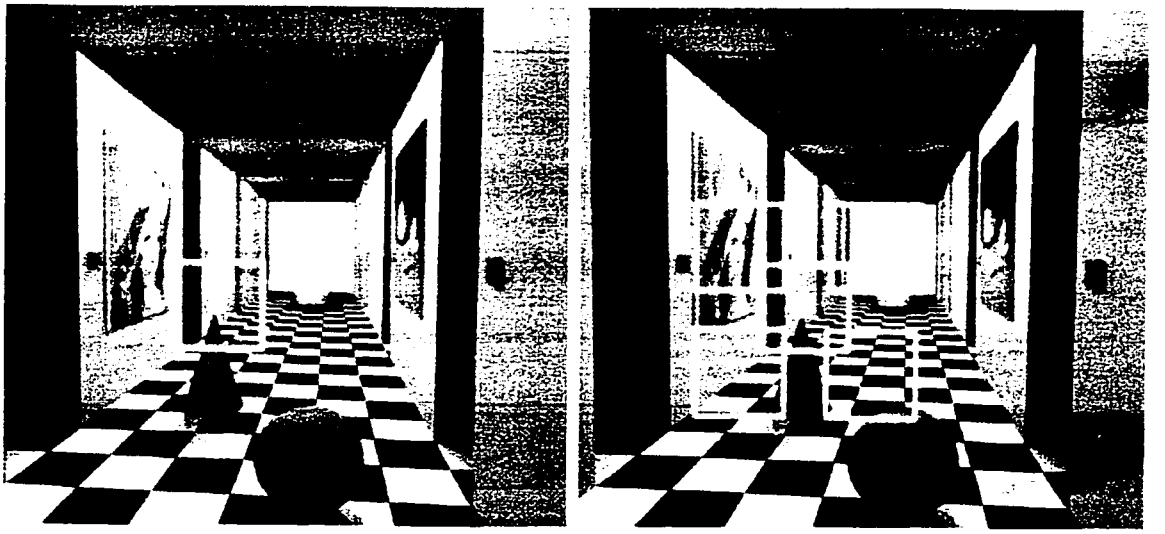


Figure 2.9: Basic of correlation techniques.

Correlation functions give a measure of the degree of similarity between two areas based on the pixels' gray level values. One of the most simple correlation functions is the Sum of Absolute Differences(SAD). SAD calculates the total absolute differences of all the pixels within the range of reference and target window. For a reference window centered around the point to be matched, the target window centered with the matching point shall produce the lowest SAD value among all other windows, which represents the highest similarity. The SAD for each pair of pixels is given by equation (2.17).

$$\mathrm{SAD}((x,y),(x',y')) = \sum_{v=0}^{vlen}\sum_{u=0}^{ulen} |R(x'+u, y'+v) - S(x+u, y+v)| \qquad (2.17)$$

A simple area-based matching algorithm can be sketched below: using f as correlation function, for each pixel $p = (x, y)$ in the left image, the following steps are carried out:

1. Within the search region of the right image, using the template of the left image associated with p to compute f.
2. Select the template of the right image that gives the maximum value for f.
3. The pixel associated with this template is the matching $p\prime$ for p.
4. Calculate the disparity.

2.4.2 Feature-based Method

A pixel's intensity value is the basic feature of an image. However, a single pixel's gray value does not provide enough information for many applications. More commonly, an image feature refers to a higher level description of an image. In [29], the image features are defined as *local, meaningful* and *d*etectable parts of the image. *Local* describes the features related to a part of the image with some special properties, and not the global properties of an image. *Meaningful* means that the features are associated with interesting scene elements via the image formation process. *Detectable* means that some algorithms exist to detect the feature. The outputs of these algorithms are called the *feature descriptors.* For example, a descriptor for the line segment feature could consist of the coordinates of the segment's central point, the segment's length and its orientation. The detection of image features can be viewed as a pre-processing stage for matching.

In feature-based methods, the matching is based on the numerical and symbolic properties of the features obtained by feature descriptors. Instead of using correlation

as similarity measurement, corresponding elements are given by the most similar feature pair using other criteria. A simple example of the similarity criterion between feature descriptors is to calculate the inverse of the average of distances between each of the properties in the descriptors, where the maximum value gives the matching.

A simple feature-based matching algorithm is described below: the input is a pair of stereo images and two corresponding sets of feature descriptors are F and $F\prime$. For each feature f from F in the left image:

1. Compute the similarity between f and all features in $F\prime$ in the search region of the right image.
2. Select the right-image feature $f\prime$ that maximizes the similarity measure.
3. Calculate the disparity of f.

2.4.3 Area-based vs. Feature-based

Area-based methods achieve a dense matching result for the stereo image pair, while feature-based methods only match a sparse set of features between the images. However, feature-based methods have the advantage of being efficient and more robust against image variations. The comparison between these two methods is shown in Table 2.1.

	Area-based Methods	Feature-based Methods
Type of image	highly textured images; images taken from slightly different view points	images rich in feature(such as in-door scenes with many straight lines).
Implementation difficulty	easy	difficult to implement and debug
Pre-process stage	not necessary	features should be extracted by the feature descriptor and matching tokens should be selected
Computation time	calculation of correlation is very expensive	comparatively faster
Sensitivity to noise	correlation methods are sensitive to lighting changes	more robust to illumination changes and other image noise
Matching result	dense disparity map	sparse disparity map of matched features

Table 2.1: Comparison of area-based methods and feature-based methods.

2.5 Conclusion

In this chapter we have introduced some basic concepts about the formation of a two-dimensional image and the geometry of a stereo vision system. In addition, we have outlined the principles of dense matching in general and presented some constraints that are sometimes used to reduce the search process for a match. In particular, the epipolar geometry was identified as one major constraint used by most matching methods. This constraint has the advantage of being available without calibrating the cameras. Camera calibration is a tedious and difficult process that is not all the time possible. Therefore, in this thesis we consider the matching problem in the general case of uncalibrated images. In the next chapter, dense matching uncalibrated images is addressed in more detail and the existing methods are reviewed.

Chapter 3

Existing Methods for Dense Matching of Uncalibrated Images

A dense disparity map is required in many stereo applications, such as 3D reconstruction and novel view synthesis. Useful as it is, dense matching remains one of the bottlenecks in stereo vision. This is due to the large amount of computation involved in dense matching, and matching accuracy is influenced by occlusion and image noise. Camera calibration can effectively help the matching process. However, under certain situations we need to avoid the overhead of the calibrating process, which means there is no information of the camera location and relative positions available. One task within this scenario is the dense matching of uncalibrated images.

Area-based correlation methods are a natural choice for dense matching uncalibrated images. Different correlation functions have been proposed to improve the performance of area-based matching. In order to reduce the expensive computation time caused by correlation techniques, geometric constraints and certain search strategies such as dynamic programming and relaxation techniques, can be applied to limit the search area. As discussed earlier, epipolar geometry can be obtained without calibrating the camera, therefore the epipolar constraint has been strongly enforced in most of the algorithms for matching uncalibrated images.

Although feature-based matching only produces a sparse set of matched points, it

has the advantage of efficiency and robustness to image noise. These characteristics allow improvement in area-based dense matching. This class of methods usually take advantage of certain types of image features to provide some knowledge of the image for the area-based matching process.

3.1 Area-based Method

3.1.1 Correlation Functions

The basic idea of area-based methods is to search for the global maximal value of the correlation function.[1] The advantage of correlation based algorithms lies in their simple and straightforward implementation. Different correlation functions are listed in table 3.1

Correlation function SSD and SAD were used in many earlier area-based matching methods. They are very intuitive methods that only compute the absolute or squared differences of the pixel's gray value within the template window. SSD and SAD are very easy to implement and perform well as long as there is no change in lighting condition. Other more complicated forms like ZSAD, NCC and ZNCC give better performance against image noise.

Methods based on only correlation functions are inherently problematic to match images with the following patterns:

- Repetitive Texture
- Lack of Texture
- Large Displacement of Cameras
- Occlusion

[1] Global maximal value refers to the value that represents the maximal similarity; it varies regarding different correlation functions. For example SSD produces a minimal value of 0 at maximal similarity, while using ZNCC it is the value of 1.

Name	Definition
sum of squared differences	$\mathbf{SSD}\left((x,y),(x',y')\right) = \sum_{v=0}^{vlen}\sum_{u=0}^{ulen}(R(x'+u,y'+v) - S(x+u,y+v))^2$
sum of absolute differences	$\mathbf{SAD}((x,y),(x',y')) = \sum_{v=0}^{vlen}\sum_{u=0}^{ulen}\lvert R(x'+u,y'+v) - S(x+u,y+v)\rvert$
zero mean sum of squared differences	$\mathbf{ZSSD}\left((x,y),(x',y')\right) = \sum_{v=0}^{vlen}\sum_{u=0}^{ulen}\left((R(x'+u,y'+v) - \overline{R}) - (S(x+u,y+v) - \overline{S})\right)^2$
zero mean sum of absolute differences	$\mathbf{ZSAD}\left((x,y),(x',y')\right) = \sum_{v=0}^{vlen}\sum_{u=0}^{ulen}\left\lvert(R(x'+u,y'+v) - \overline{R}) - (S(x+u,y+v) - \overline{S})\right\rvert^2$
cross correlation	$\mathbf{CC}\left((x,y),(x',y')\right) = \sum_{v=0}^{vlen}\sum_{u=0}^{ulen} R(x'+u,y'+v)\cdot S(x+u,y+v)$
normalized cross correlation	$\mathbf{NCC}\left((x,y),(x',y')\right) = \frac{\sum_{v=0}^{vlen}\sum_{u=0}^{ulen} R(x'+u,y'+v)\cdot S(x+u,y+v)}{\sqrt{\sum_{v=0}^{vlen}\sum_{u=0}^{ulen} R(x'+u,y'+v)\cdot\sum_{v=0}^{vlen}\sum_{u=0}^{ulen} S(x+u,y+v)}}$
zero mean normalized cross correlation	$\mathbf{ZNCC}\left((x,y),(x',y')\right) = \frac{\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}(R(x'+u,y'+v)-\overline{R})\cdot(S(x+u,y+v)-\overline{S})}{\sqrt{\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}(R(x'+u,y'+v)-\overline{R})^2\cdot\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}(S(x+u,y+v)-\overline{S})^2}}$

Table 3.1: Definition of correlation functions (R and S denote the images; *ulen* and *vlen* are the size of correlation window).

If the image texture is repeated, multiple possible correspondences may exist; in the case of occlusion, false matches may be found. Matching results also become very sensitive when the images are taken from very different viewpoints. Furthermore, the correlation functions are not guaranteed to provide a unique global maximal. The correct match may correspond to a value other than the maximal of the correlation function. When there is very little texture in a scene, the similarity function may have a wide plateau of indistinguishable maxima, which results in ambiguous matches. Therefore, we need to include more constraints and adopt matching strategies to improve the result of simple area-based matching methods.

3.1.2 Disambiguate Multiple Match

Ambiguous matches are caused by applying correlation functions on images with certain texture patterns, where more than one pixel in one image are matched to the same pixel in the other image. We can reduce the impact of this problem by enforcing the epipolar constraint. Note that in addition, the use of the epipolar constraint also reduces the size of the search region. A threshold can also be used to eliminate unlikely matches, and the left-right coherence is another way of verifying matching results.

Constraint and Threshold

Looking for the global maxima of the correlation function in the whole image is a very naive and impractical approach. Thus, geometric constraints for matching have been exploited in many methods to help in restricting the search area and discarding false matches. The search area can be effectively limited by applying the epipolar constraint. As we have discussed, the search over a 2D area is restricted to a 1D epipolar line when using the epipolar constraint. This enables a more efficient and accurate matching process. Also, the order constraint can be used to discard impossible areas in which the match may occur.

The threshold is a common way for controlling reliability in matching. We can set a threshold value for similarity functions so that all the values lower than this threshold are discarded. However, the threshold value is arbitrary and is hard to choose in

practice. Other statistic methods can be applied to qualify matching candidates. One example is the method of Krotkov *et al.* [19] which requires the match to have a maximal value of the correlation function that is higher than any other value by some percentage.

Left-right Coherence

The left-right coherence check relies on the following fact: for a pixel p in the left image with its match $p\prime$ in the right image, the disparity for p and the disparity for $p\prime$ should be equal in absolute value but have opposite algebraic signs. Based on this requirement, Hannah [14] used a strategy to refine the result of disparities by horizontally flipping the left and right images. The results are then re-analyzed by using the flipped left image as the new right image and the flipped right image as the new left image. Similarly, Fua [11] used two images symmetrically to validate the result. Only consistent matches are accepted, such that if p in left image matches to $p\prime$ in right image, then $p\prime$ must match to p when performing a reverse matching process. This check can be used to eliminate multiple matches; however it becomes problematic at object boundaries where occlusion is most likely to occur. It is also a costly process because the computation time will be doubled. Therefore, the left-right coherence check can be applied to only a limited number of cases.

3.1.3 Using Template Windows with Variable Sizes

Classical correlation matching methods are based on comparing windows of a fixed size. The selection of the window size is an important issue since the window size must be large enough to include enough intensity variation for matching, but also small enough to avoid the effects of projective distortion. If the window is too small, then many false matches will be found; for example, just one pixel will have hundreds of matches (same color pixels). If the window is too large then different regions of different disparity will be combined together, and there will be no real good match found. Some methods have been developed which select the window size adaptively .

Levine *et al.* [21] proposed a primitive method for using window size which is locally

depended on the intensity pattern. Based on his work, Kanade [18] proposed a more robust and accurate method that uses adaptive window based on the local variation of intensity and disparity. This algorithm starts with an initial estimate of the disparity map. For each pixel, an uncertainty measurement is also adopted as local support. Based on the information of disparity estimate and uncertainty value, the size of the correlation window can be adjusted until the optimal size is reached. However, this method requires the model of the disparity map to be available in advance. In addition, the determination of the appropriate window size for each pixel is a difficult and time consuming task. Saito [26] employs several disparity maps that are generated by SSD correlation using different window sizes. Then the optimal disparity map is determined by combining these disparity maps using a genetic algorithm. The result is satisfactory but the multiple matching process makes this method very impractical.

3.1.4 Dynamic Programming

Dynamic programming is an algorithmic approach to solving problems by effectively using the solutions to sub-problems. Progressively larger problems are solved by using the solutions to sub-problems, thus avoiding redundant calculations. This strategy is applied when an intrinsic ordering of the problems exists. Since the order preserves along the corresponding epipolar lines, the pair of epipolar lines are good candidates for scanlines to apply dynamic programming in stereo matching.

Figure 3.1 is a graphical representation of the stereo matching problem. In order to use dynamic programming to solve the matching problem, each vertex is labeled using the array $P(i, j)$. Index i and j refer to pixels on left and right epipolar lines respectively. The diagonal edges are solid edges, which represent a possible matched pair. Solid edges are associated with the correlation score of the appropriate pixel pair. Lower scores represent high similarity for correlations such as SSD or SAD. The horizontal edges are dashed edges, which represent occlusions in the disparity map. A dashed edge is given an occlusion cost of C_o, which allows for occlusion to occur. The matching problem can be viewed as finding the shortest path between the vertices labeled S and T.

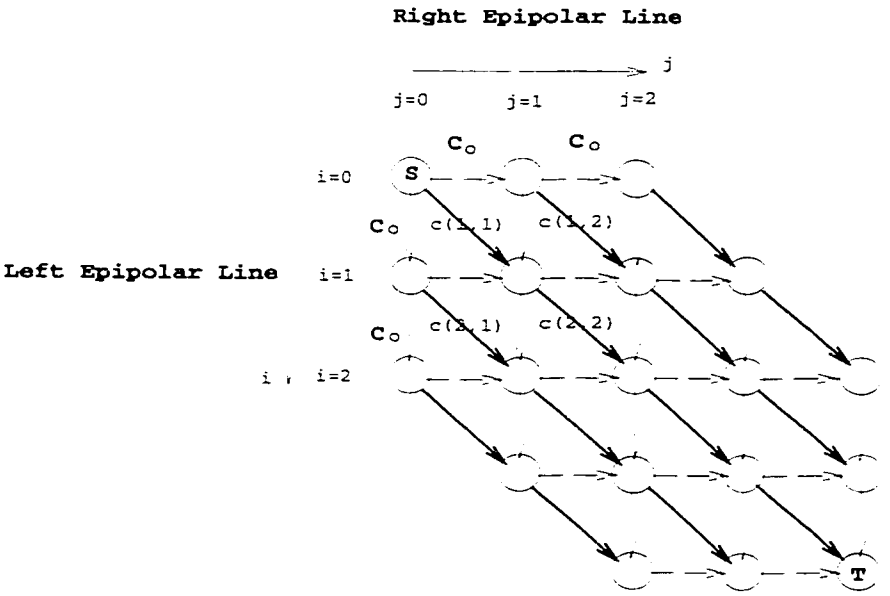


Figure 3.1: Using dynamic programming to solve stereo matching.

Different methods using dynamic programming have been designed. Ohta [25] proposed to enforce the continuity constraint in the cost, and he tried to find the path in a 3D space instead of the previous 2D space. His improvement is at the cost of a large amount of computation. Lloyd's method [22] consists of two stages: first a set of candidate matches are produced for each row, then using continuity constraints choose the best among the candidates.

In dynamic programming solutions, the choice of the occlusion cost is of critical importance. If C_o is too low the result will be consist of occlusion path only, while a very large C_o will result in no occlusions. Also, the results often exhibit spurious vertical discontinuities in disparity, since the monotonic ordering constraint is not satisfied when narrow objects exist [32]. Moreover, it is very costly in computation time to use dynamic programming to achieve dense matching for a stereo pair of images.

3.1.5 Relaxation Technique

Relaxation techniques in stereo matching resemble the iterative numerical relaxation methods. They use a bottom-up search strategy that involves local rating of similarity which depends on the ratings of the neighbours. These ratings are updated iteratively until the ratings converge or until a sufficiently good match is found. In other words, first an educated guess (what the matching should be) is produced, then the match is reorganized by propagating some of the constraints. There are two types of approaches for relaxation techniques: probabilistic-based and optimization-based.

In the probabilistic-based approach, the initial probabilities are computed from similarities in the values surrounding the match points. Then these probabilities are updated iteratively to impose global consistency. These iterative procedures are repeated until either the probabilities reach a steady state or a certain termination condition is satisfied. In each iteration of the relaxation process, the probability value must be updated according to the current probability value and the neighbour's information [3]. Probabilistic-based methods require a lot of processing time while trying to find the global transformation. This kind of method becomes impractical since in most cases of matching uncalibrated images global transformations can not be found.

In the optimization-based approach, matching is carried out by minimizing an energy function where energy functions are formulated from the constraints. It represents a mechanism for the propagation of constraints among neighbouring match points where the multiple-match ambiguity can be removed iteratively. An example of this type of method is proposed by Marr [23], where the uniqueness and the continuity constraints are enforced. We denote points of the first image by l_i, and points of the second image by r_j. For each pixel l_i of the first image, we compute an initial set of confidence measures $c_0(l_i, r_j)$ that estimate whether l_i matches r_j in the second image. r_j are chosen on the epipolar line of l_i, thus the epipolar constraint is imposed. There are several ways of computing $c(l_i, r_j)$, the simplest one is by comparing their intensity values.

$$c_0(l_i, r_j) = \begin{cases} 1 & \text{if the intensity at } l_i \text{ in first image is similar enough to the intensity at } r_j \text{ in the second image.} \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Then the confidence measures are updated according to the following function:

$$c_{n+1}(l_i, r_j) = \begin{cases} 1 & \text{if } k \text{ is above some threshold; } k \text{ represents the number of pixels } l_i\prime \text{ in a neighbourhood of } l_i, \text{ such that } c_n(l_i, r_j) = 1 \text{ for } r_j\prime \text{ in a neighbourhood of } r_j. \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

The idea underlying equation (3.2) is to enforce the continuity and uniqueness constraint around the matching neighbourhood.

Marr's method is tested only on random-dot images; it is very sensitive to the settings of the threshold and other parameters. Several other proposed energy functions require the knowledge of camera parameters. For uncalibrated images, Yokoya [31] proposed an energy function that combines a similarity term and a smoothness term. A dense disparity map is computed by solving a system of equations using a coarse-to-fine approach. [12] [28] [10] are methods that aim at overcoming the problem of smoothness over edges caused by the relaxation process, but at enormous computational costs. Also some other new energy functions are proposed; see for instance [1][6].

The problem of relaxation based approaches is that they do not always converge and do not always recover the correct matches. Moreover, the minimizing of some energy functions is NP hard. In general, the implementation using relaxation is very complicated and the iterative calculation produces a heavy burden on the CPU re-

source.

3.2 Hybrid Methods: Combining Dense and Sparse Matching

Using area-based methods to achieve dense matching usually suffers from the drawbacks of the heavy computation cost. On the other hand, feature-based methods only give a sparse matching result of a set of feature points, but at a relatively much lower cost. Research has been done towards fusing these two techniques resulting in hybrid methods. Hybrid methods take advantage of the unique attributes of each of these techniques: the area-based process provides a dense disparity map and the feature-based process provides a reliable and accurate match for a limited number of pixels, or interest points. The basic idea behind combining the two techniques is to integrate the image feature information in area-based matching to produce a more efficient and accurate result.

Typical interest points are corners, line intersections and object boundaries. The selection of features to be used lays the foundation for the matching accuracy. In general, the chosen features should be tolerant of local distortions and the number of features should be sufficient to perform the calculation, but not too large in order to ensure the efficiency of feature matching.

3.2.1 Early Research

A matching approach that takes into account feature information is first proposed by Barnard [4]. In this method, a set of candidate matching points are selected independently in each image at the beginning. These points are local feature points such as spots and corners. After two sets of candidate points are found, possible matches are then constructed based on SSD correlations. Then, a probabilistic-based relaxation process is carried out to refine the result. Although this method does not aim at achieving a dense matching result, it raises the idea of using feature information other

than only similarity measurement. The experiments have proven that this method is less sensitive to noise and distortions.

As a modified algorithm of Marr's method [24], Grimson [13] proposed an algorithm that is characterized by matching certain symbolic features in filtered images. The features obtained by a zero-crossing filter indicate the locations of significant changes in intensity. These feature points consist of two sets: points that tend to form extended contours and points that lie scattered in small segments. Object contours can be obtained by resolving these two kinds of features. At the end, an interpolation process is applied to the object surfaces based on the object contours. This method is mainly tested on aerial images to obtain a terrain contour map. The problem of this method is that it does not give an accurate dense match for all points especially when the scene has more depth changes. This is because the feature points used are not clearly defined corners or lines that can represent an accurate location of discontinuity.

Hoff and Ahuja [17] proposed a method that integrates feature matching, contour detection and surface interpolation. In their method, an initial coarse estimate of the disparity map is used to predict the search area for matching. This coarse estimate is done by comparing big windows of the correlation template. Then, edges are extracted by an edge operator. The matching process is an integration of matching and interpolation based on the edge information. Matching and interpolation are done by fitting planar patches and finding the occlusion and ridge contours. Fitting planar patches can be viewed as the matching of all points in a small local area. Starting from coarse, the edge operator successively changes the parameters and, the matching and interpolation can be performed until the final result is reached. The interpolation process based on edges is combined with the matching process in order to get a dense disparity map. However, this method is difficult to implement and computationally very expensive.

3.2.2 Later Research

Cochran and Medioni [8] proposed a method which consists of several processing stages. In the initial process, noise is removed from the original images and the images are

aligned (parallel epipolar lines). Then, an area-based matching process that produces an initial estimate of dense disparity map is carried out using normalized cross correlation(NCC). During the matching, the order constraint is enforced as well as the right-left coherence rule. This allows the elimination of unreliable matches. Finally, a refinement based on the edge information of the images is performed. After the previous three steps, the interpolation is also used to obtain a dense disparity map without gaps.

Weng *et al.* [30] described a matching approach that uses multiple attributes associated with each pixel that yields a general system of constraints. Pixel intensity, edge, and corner features are used in this method to provide an over-determination for matching. Edge and corner features are blurred to different resolution levels to provide information needed for matching. Matching is done in a coarse to fine style in order to cope with large disparity and achieve refined results. This approach is mainly aimed at matching nonrigid scenes where the epipolar geometry does not hold. Thus, the epipolar constraint is not enforced.

Kumar and Desai [20] developed an approach which integrates three modules: feature extractor, matching and interpolation module. The Edges, which are extracted from two images using an edge extractor, are matched by applying an energy function. A multi-resolution approach is adopted to perform the matching at different levels of the image hierarchy interactively. The final dense disparity map is obtained after the interpolation process. To preserve discontinuities on the surface, line fields are incorporated too. This approach is only tested on aerial images, and due to the use of energy function, the computational cost is high.

3.3 Conclusion

Since area-based matching using correlation functions does not require information about camera parameters, it is a natural choice for dense matching of uncalibrated images. However, computing correlation functions is very expensive, which effects the efficiency of most area-based algorithms. Unlike most classical dense matching approaches, hybrid methods that integrate the area-based and feature-based primitives

are more reliable and more efficient. The hybrid matching process takes into account the feature information which is actually an important prerequisite in order to get reliable and accurate dense matching results. However, most existing hybrid methods proposed are not practical and difficult to implement. Therefore, one of our goals in this thesis is to design an efficient and accurate hybrid dense matching algorithm. In particular, the proposed algorithm should be easy and straightforward to implement. Also the computational and hardware requirements should be kept at a minimum level.

Chapter 4

Experimental Study of Dense Matching for Uncalibrated Images

In this chapter, we first describe a simple approach for the dense matching of uncalibrated images. In this approach, an exhaustive search over the epipolar line is constructed to find the best correlation score. The correlation function we have used is ZNCC, given its robustness and stability. Then, a modified algorithm using image interest points is proposed. Image interest points provide a disparity estimate for the matching process of all the remaining points. This improved dense matching algorithm is faster since the search area for each pixel is reduced. Experimental results show the improvement of CPU time and matching accuracy.

4.1 Exhaustive Search

In this approach, the search for the match is exhaustively carried out on the corresponding epipolar line in the second image. In our case, we have used the correlation function ZNCC. As discussed earlier, applying the epipolar constraint in matching can effectively reduce the search from a 2D image to a 1D epipolar line. To find a match in the right image for a pixel p of the left image, we consider all pixels on the corresponding epipolar line of p in the right image as p's candidate matches. The ZNCC of

p and each candidate is calculated and the pixel associated with the highest score is taken as the match.

4.1.1 Correlation Function ZNCC

ZNCC stands for Zero-mean Normalized Cross-Correlation; it is a more robust correlation function than traditional ones, such as SSD, SAD and CC. ZNCC is defined as follows:

$$\mathbf{ZNCC}\left((x,y),(x',y')\right) = \frac{\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}(R(x'+u,y'+v)-\overline{R})\cdot(S(x+u,y+v)-\overline{S})}{\sqrt{\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}(R(x'+u,y'+v)-\overline{R})^2\cdot\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}(S(x+u,y+v)-\overline{S})^2}} \quad (4.1)$$

where R and S are the template window, $ulen$ and $vlen$ denote the window size.

ZNCC is evolved from the cross correlation(CC) function. CC can be normalized using the local image energy $\sum\sum R(u,v)\cdot\sum\sum S(u,v)$, in order to be more robust to local image intensity variation. The statistical measure of adding mean deviation can then be applied for NCC, which results in ZNCC. ZNCC measures correlation on an absolute scale range of [-1, 1]; thus it is advantageous compared to other relative measurements. The work of Aschwanden [2] shows that ZNCC is very robust against many types of image distortion and noise. Therefore, we choose ZNCC as the correlation function in our work.

To avoid the additional computational cost introduced, the ZNCC formula needs to be optimized. From equation (4.1), one can note that the denominator is the multiplication and square root of two parts:

$$\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}\left(R(x'+u,y'+v)-\overline{R}\right)^2 \quad (4.2)$$

and

$$\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}\left(S(x+u,y+v)-\overline{S}\right)^2 \tag{4.3}$$

These parts compute the square difference of a pixel and its associated template's average value $\overline{R}$ and $\overline{S}$. Here $\overline{R}$ denotes the average intensity of the target window of a candidate match, and $\overline{S(x,y)}$ the average intensity of the reference window of the pixel to be matched. To avoid multiple calculations of these averages caused by the sum operation, we can rewrite (4.2), assuming window length and width to be $vlen = ulen = n$.

$$\begin{aligned}
&\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}\left(R(x\prime+u,y\prime+v)-\overline{R}\right)^2\\
=&\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}\left(R^2(x\prime+u,y\prime+v)+\left(\overline{R}\right)^2-2R(x\prime+u,y\prime+v)\cdot\overline{R}\right)\\
=&\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}R^2(x\prime+u,y\prime+v)+n^2\left(\overline{R}\right)^2-2\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}\left(R(x\prime+u,y\prime+v)\cdot\overline{R}\right)\\
=&\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}R^2(x\prime+u,y\prime+v)+n^2\left(\overline{R}\right)^2-2n\cdot\overline{R}\cdot\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}R(x\prime+u,y\prime+v)\\
=&\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}R^2(x\prime+u,y\prime+v)+n^2\left(\overline{R}\right)^2-2n\cdot\overline{R}\cdot n\cdot\overline{R}\\
=&\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}R^2(x\prime+u,y\prime+v)-n^2\left(\overline{R}\right)^2
\end{aligned} \tag{4.4}$$

similarly, we can rewrite (4.3):

$$\begin{aligned}
&\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}\left(S(x+u,y+v)-\overline{S}\right)^2\\
=&\sum_{v=0}^{vlen}\sum_{u=0}^{ulen}S(x+u,y+v)-n^2\left(\overline{S}\right)^2
\end{aligned} \tag{4.5}$$

The numerator of equation (4.1) can also be rewritten as follows:

$$\begin{aligned} & \sum_{v=0}^{vlen} \sum_{u=0}^{ulen} (R(x'+u, y'+v) - \overline{R}) \cdot (S(x+u, y+v) - \overline{S}) \\ = & \sum_{v=0}^{vlen} \sum_{u=0}^{ulen} R(x'+u, y'+v) S(x+u, y+v) - n^2 \cdot \overline{S} \cdot \overline{R} \end{aligned} \tag{4.6}$$

Now, we can use formulas (4.4), (4.5) and (4.6), where the calculation of averages are not related to the sum operation. The averages are calculated only once, and can be retrieved each time needed. Thus, the total computation cost of ZNCC is considerably reduced.

4.1.2 Epipolar Line

Since the epipolar constraint is strongly exploited in this approach and our next proposed algorithm, it is important to correctly calculate the corresponding epipolar line for a given pixel. There are two tasks involved: calculating the coefficients of the line and scanning the epipolar line on the target image.

For uncalibrated images, there is no known information on the cameras' parameters or their relative positions and orientations. However, the fundamental matrix F can be calculated from a few matches in a pair of images. Given a pair of uncalibrated images, F is calculated in our experiments using a program provided by Zhang [33].

As we have mentioned earlier, given a pixel p in the left image(reference image) and F between the left image and right image(target image), the coefficient of the corresponding epipolar line can be given by

$$F \cdot p = (a, b, c)$$

where the equation of this line is $ax + by + c = 0$.

To plot the line in an image of pixel coordinates, a line drawing algorithm should be used. We have used Bresenham's line algorithm to determine the pixel positions on the line.

Bresenham's algorithm plots a line between a start point and an end point. A parameter is defined in this algorithm to provide a measure of the relative distances of two pixels from the actual position on a given line. Based on this parameter, the closest pixel to the line is chosen. Detailed explanation of this algorithm can be found in [16].

4.2 Integrating Interest Points

In this section, we have modified the simple approach of using correlation and exhaustive searching by taking into account image interest points. Interest points are some feature points, mostly corner points that can be matched very precisely between two images. Matched interest points are the pair of interest points with established correspondence. For the sake of simplicity, we shall refer to the matched pair of interest points as interest points.

Since these interest points reflect somewhat a prior knowledge about the disparity change around their neighbourhood, we can use this information to estimate the disparity of their neighbour points. Thus, the search area for the matching process can be reduced based on this estimation.

4.2.1 Interest Points

To apply the disparity estimation, first we need to decide the interest points. The method we have used for extracting and matching interest points is based on the work of Zhang [33]. He proposed a robust technique for estimating epipolar geometry by sparse matching two uncalibrated images. In his method, corners in two images are first extracted as feature points. Then using the correlation technique, these corner points are matched, followed by a robust statistic method which is applied to discard false matches within this set of matched points. Later, a relaxation process is performed to refine the matches. These resulting matches are the basis for estimating epipolar geometry.

We adopt the initial set of matched corner points obtained in Zhang's method as our

interest points. An example set of interest points is shown in Figure 4.1. In his work, this set of points are used to calculate the epipolar geometry. Although the matching result is of a very good quality, it is impossible to apply the same strategy to the whole image due to the computational cost. In our work, these points are integrated into the dense matching process to act as a disparity estimate for local areas. Obtaining interest points can be viewed as a pre-processing stage of the dense matching.

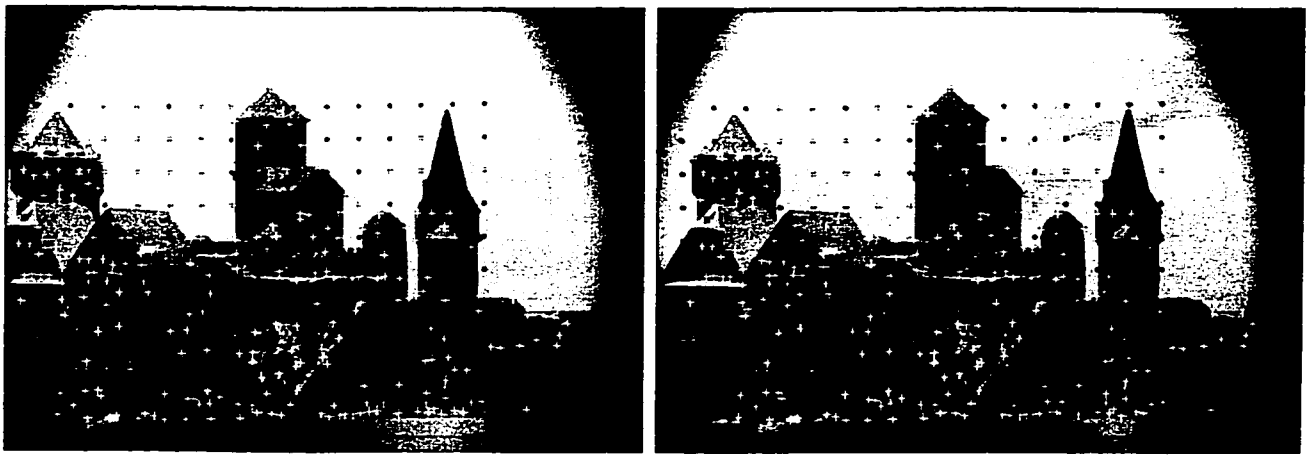


Figure 4.1: An example of matched interest points.

4.2.2 Disparity Estimate

The disparity estimate is based on the disparity of the matched interest points. For a certain pixel to be matched, instead of searching through the entire epipolar line for the candidate match, we can limit the search area using the disparity information of the closest interest points. In Figure 4.2, p is the pixel point to be matched in the left image, and I is an interest point that has already been matched to $I\prime$ in the right image. The disparity between I and $I\prime$ gives a disparity estimate for the pixel p. Since p and I are in the same neighbourhood, this disparity estimate is a very good guess for the disparity between p and $p\prime$. Therefore, we can roughly locate p' in the right image. The rough location of p' is marked with a question mark in Figure 4.2. Hence, the search for the actual p' can be conducted in the local area around this rough location instead of the whole epipolar line.

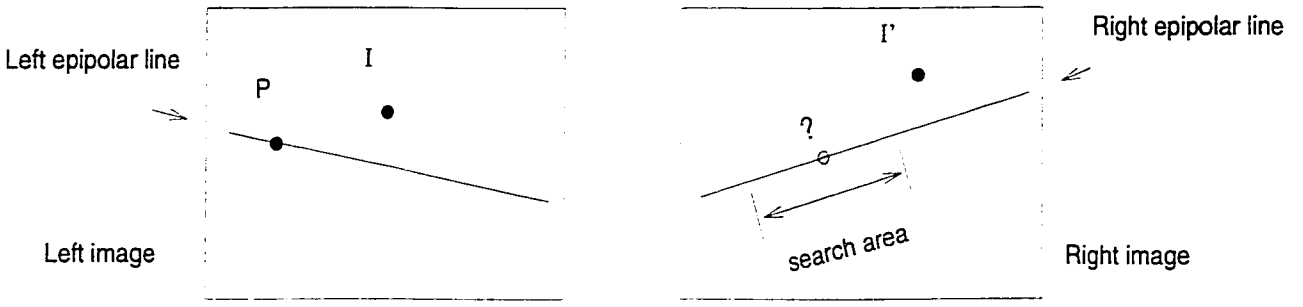


Figure 4.2: Limiting the search area based on interest points.

For a pixel p from the left image to be matched in the right image, the task is to choose a pair of matched interest points and use their disparity as a guess. The most intuitive way is to use the disparity of the closest interest point to p in the left image as a disparity guess for p. The number of interest points obtained is usually large enough. Also, interest points are mostly well spread in the image, providing a reliable estimate for their neighbouring pixels.

However, when the image size is large and a large number of interest points are available, it is not efficient to calculate all the distances between p and all the interest points. Instead of these exhaustive calculations, we can define a grid to reduce computations. As shown in Figure 4.3, the search for the closest interest point to p in the left image can be carried out within the grid (sub-image) in which p is located.

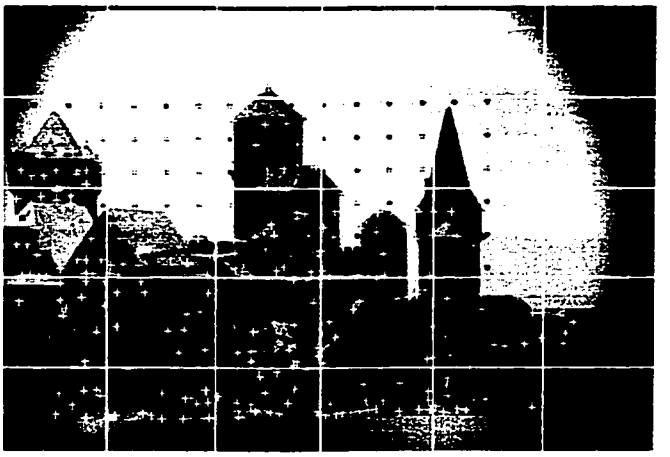


Figure 4.3: Using grid to search for the closest interest point.

Figure 4.4: Stereo image pair: Head(size 384 × 288).

4.3 Experimental Results

We have tested the method of exhaustive search and the one based on an integration of interest points on two different scenes: Castle scene(Figure 1.2) and Head scene(Figure 4.4). In our experiments, the matching is carried out from the left to right image. To make it convenient for viewing, we display the reconstructed right image from the matching result as the output. The pixels in the output image is by default white(intensity 255), before matching occurs.

Results of Exhaustive Search

The result of the exhaustive search method is shown in Figure 4.5. The first row of the figures shows two reconstructed images using different correlation window sizes for the same Castle scene. The second row of the figures shows the result for the Head Scene. Although there is a slight improvement when using a 9×9 window over a 7×7 window, the overall matching quality is worse than we would expect. A lot of white spots appeared due to pixels that were not matched. On the other hand, many spots were not correctly recovered because the matching of these pixels was wrong. The number of mismatches is so high that the original objects can hardly be recognized. This indicates that even with the choice of a robust correlation function ZNCC, applying only the epipolar constraint is not enough for stereo matching. It proves that the correct match for a pixel may not correspond to a global maximal correlation value along the epipolar line.

Moreover, the computation time was very high. As shown in Table 4.1, it takes almost one hour and a half to match a stereo pair of the Castle scene. using a window of size 9×9. Obviously, we need to adopt other search strategies and constraints besides the epipolar constraint for improvement.

Results of Integrating Interest Points

Figure 4.6 and Figure 4.7 show the results when using interest points as disparity estimates. The closest interest point is located by the exhaustive search technique.

The grid search method is not a big advantage in this case, since the amount of total interest points is comparatively small due to the size of the image. To look for the matching point, the search area is set to 5 pixels at each side of the guessed match.

The results have considerably improved with respect to both matching accuracy and efficiency. Compared to the exhaustive search using the same size correlation window, this approach reduces unmatched pixels and mismatches in great numbers. The computation time is also reduced more than 40 times for Castle scene and 25 times for the Head scene.

Again, the quality of the reconstructed image is improved using a bigger correlation window. Better results are obtained at the cost of more time. This is because a bigger window includes more information for matching, while more calculations are involved. However, many unmatched points still persist even in the matching by correlation window 13×13. To further improve the result, and reduce computation time, we need to exploit other matching constraints.

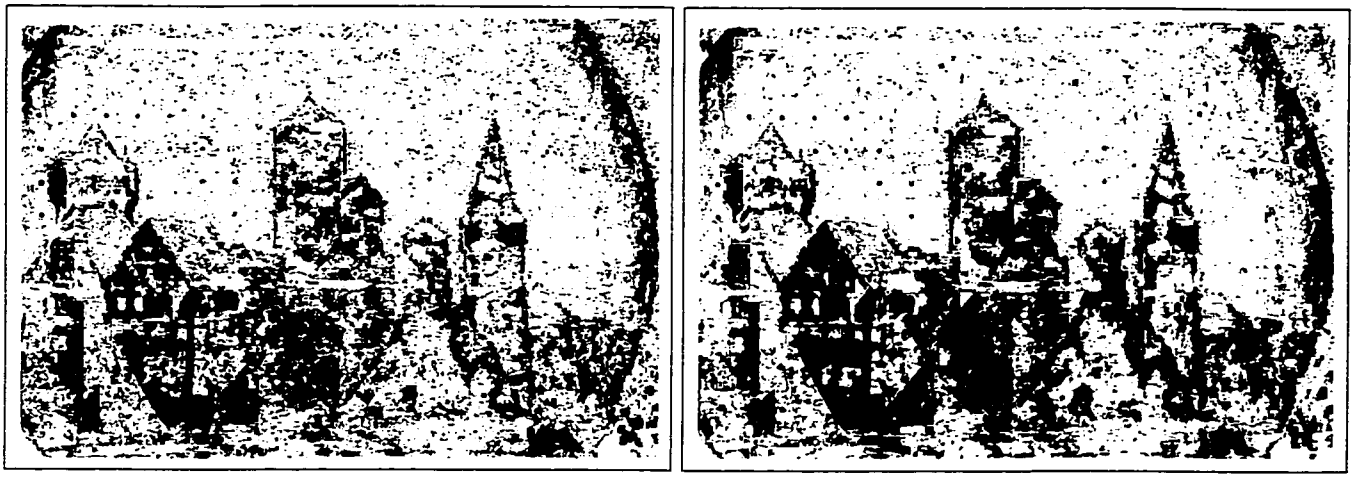
Exhaustive search

	correlation window size			
	7×7	9×9	11×11	13×13
Castle scene	111s	156s	210s	270s
Head scene	50s	70s	95s	124s

Integrating interest points

	correlation window size	
	7×7	9×9
Castle scene	3373s	5338s
Head scene	1097s	1730s

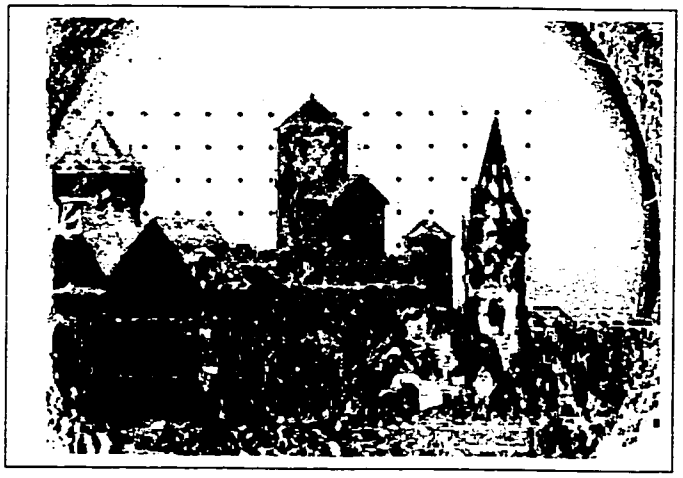
Table 4.1: Processing time of exhaustive search approach and interest points approach.



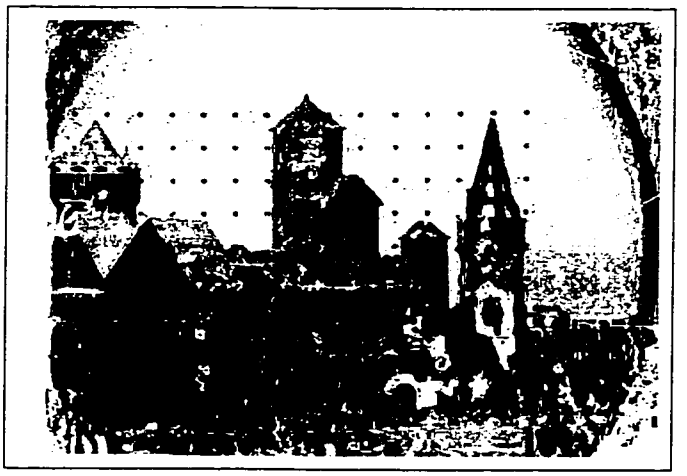
correlation window 7×7 *correlation window* 9×9

correlation window 7×7 *correlation window* 9×9

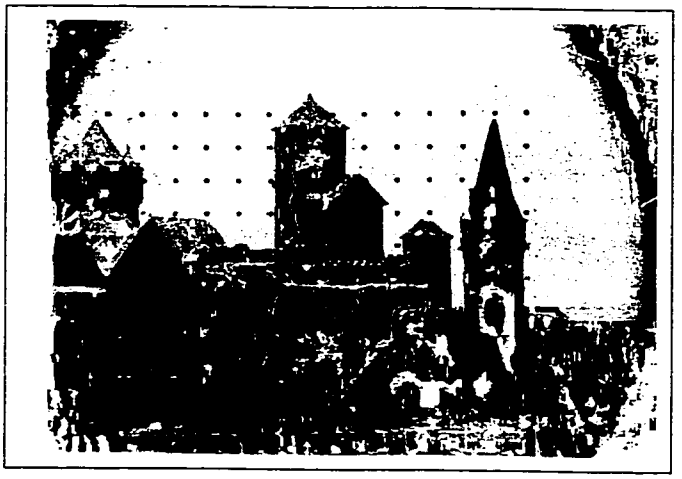
Figure 4.5: Matching results of exhaustive search algorithm for the Castle scene and the Head scene.



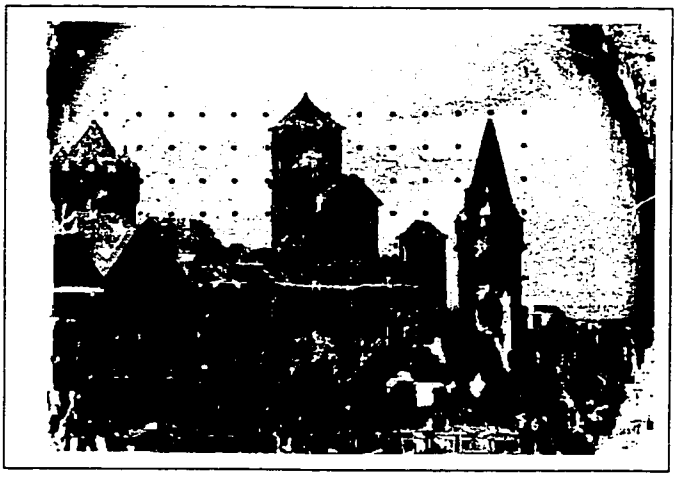
correlation window 7×7



correlation window 9×9

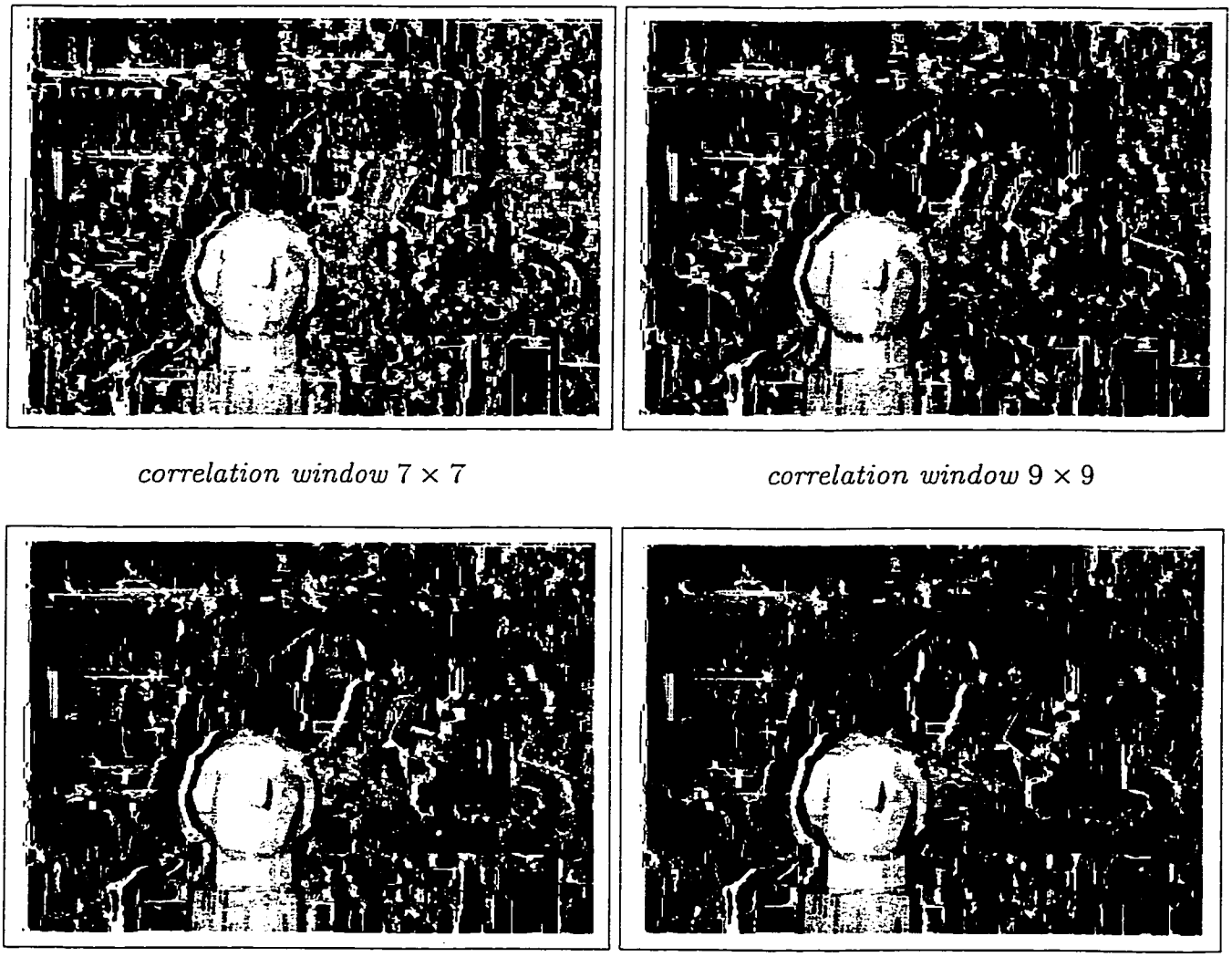


correlation window 11×11



correlation window 13×13

Figure 4.6: Matching results of interest points algorithm for Castle scene.



correlation window 7×7 *correlation window* 9×9

correlation window 11×11 *correlation window* 13×13

Figure 4.7: Matching results of interest points algorithm for Head scene.

4.4 Conclusion

In this chapter, we studied a simple approach of exhaustive search based on the correlation function ZNCC and epipolar constraint. The experiments have shown that the time for exhaustive search was very high, while the matching accuracy was very poor. We have proposed an improved version to the simple exhaustive search algorithm that takes advantage of the interest points. Image interest points provide disparity estimates for every pixel, making the matching process more efficient by reducing the search area for candidate matches. In addition, the likelihood of a mismatch is greatly reduced because the search is conducted in a small area. The experimental results have shown significant improvements in CPU time and matching quality. However, the interest points do not provide enough information for the whole image structure. Furthermore, the accuracy of the object boundaries is not reliable. We need to include more reliable image features in the matching process. In the next chapter, we propose an approach that integrates image edge features to obtain a faster and more accurate algorithm.

Chapter 5

Toward a Fast Dense Matching Algorithm

In this chapter, we present a fast dense matching algorithm which integrates the edge features of images. As discussed in the previous chapters, one major problem of the area-based dense matching algorithm is its high computational time. Computing correlation functions for candidate matches is very costly. Some methods have attempted to integrate image feature information in the dense matching of uncalibrated images. However, most of them are not practical and difficult to implement. In Chapter 4, we proposed an approach where interest points were used for disparity estimates. Although the matching quality is improved, interest points are primitive features that do not provide enough information of the image structure. Therefore, here our intention is to design a hybrid matching algorithm that achieves the following:

- Preserves disparity discontinuity at object boundaries.
- Improves time complexity.
- Enables simple and straightforward implementation.

5.1 Integrating Edge Feature

As we have pointed out, the disparity guess based on image interest points is not enough. For instance, given a point at an object boundary, the closest interest point found for this point may be actually located on a different object with a very different depth in the scene. The search region that will be guessed based on this interest point's disparity may not include the correct match.

Therefore, we consider using another important type of image feature: edges. Edges have advantages over interest points since the borders of objects in a scene all generate edges. The knowledge of the edge information allows us to locate the area where possible sharp disparity changes may occur. By segmenting the image into edge and non-edge areas, we can apply different matching strategies over these two types of areas for optimal results.

A flow diagram of our approach is shown in Figure 5.1. First, edges are extracted from the left image, and the image is divided into two parts: edge part and non-edge parts. Then we match all pixels that belong to the edges of the left image to their corresponding pixels in the right image using the interest point algorithm. To match the non-edge area, we use another algorithm which exploits most of the geometrical matching constraints. According to continuity constraint, at the non-edge areas discontinuity changes are smooth. Therefore, the search for the candidate match is limited to a restricted area. A simple interpolation process is also applied for refinement. The final dense matching result is achieved by combining the matching of edge and non-edge part.

5.2 Matching Edge Area

5.2.1 Edge Detection

In general, edge detection involves three steps. We smooth the image first to reduce the noise; then some derivative operators are applied to the image and finally the edges from the output of the derivative operator are labeled. Canny [7] introduced the

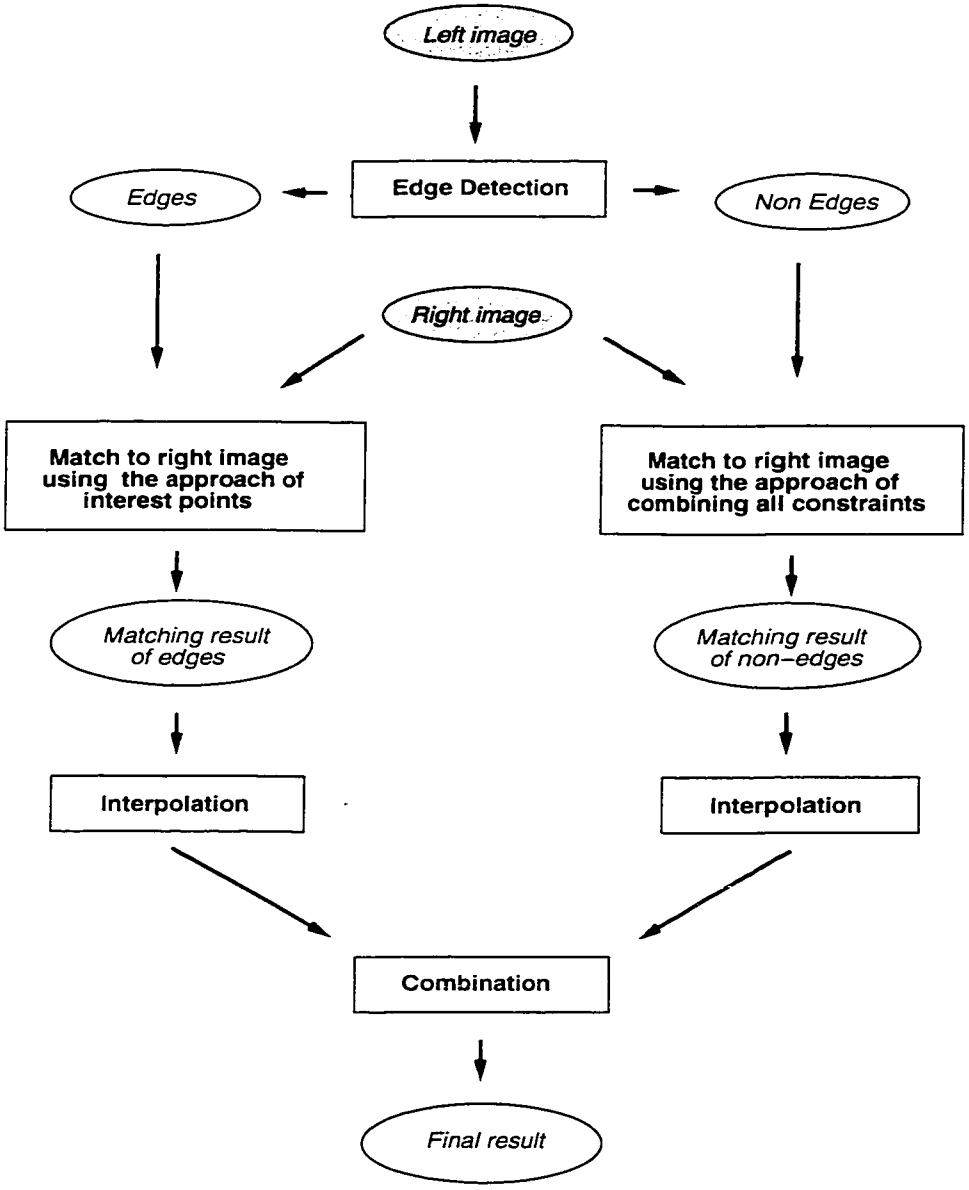


Figure 5.1: The flow diagram of our hybrid approach.

mathematical criteria for an optimal edge detecting filter: a good algorithm should detect as much as the real edges in the image and the detected edge should be as close to the true edge as possible. Deriche [9] used Canny's criteria to derive a more efficient solution. In our work, we have a Deriche's edge detector, downloaded from the INRIA web site, to extract edges from the images.

Note that the edge area we use in our work is not the direct output of Deriche's edge detector, but a small neighborhood of several pixels in addition to the original edges. This is to ensure that all boundary pixels, even the ones missed by the edge detector, are included. It is safer to label a non-edge pixel as an edge pixel than the other way around. Figure 5.2, 5.3 and 5.4 show the left image of the image pair Castle, its output of the Deriche edge detector, and the edge area we extracted from the image(including the neighboring 5 pixels). The time cost for detecting edge areas is negligible compared to the amount of time for the whole matching process.

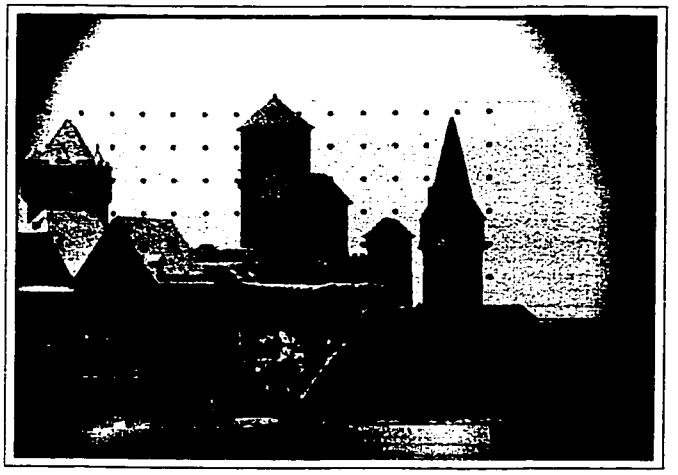


Figure 5.2: The left image of the stereo pair Castle.

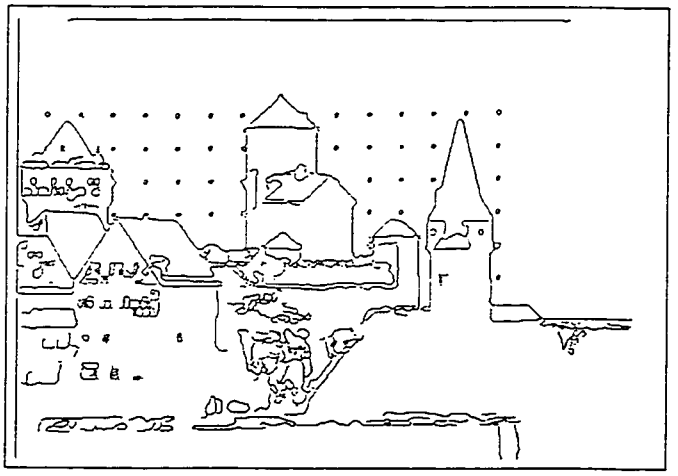


Figure 5.3: Output of the Deriche edge detector.

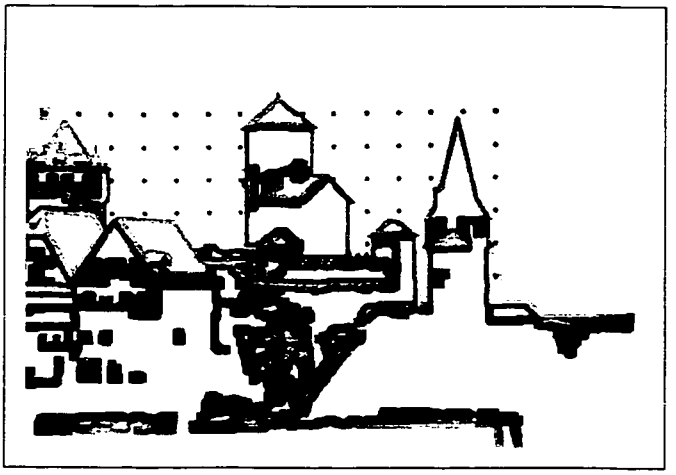


Figure 5.4: Edge areas extracted from the original image.

5.2.2 Matching Edge Area

The dense matching algorithm based on interest points, proposed in Chapter 4, is used as a base for matching edge area. The reason for choosing an area-based approach rather than a feature-based approach is manifold. First, comparing to feature-based method that extracts and matches feature descriptors, an area-based method is much easier to implement. In addition, the edge map obtained is difficult to match to the other image using a feature-based method. Second, since the edge area represents only a small area in the whole image, an area-based method will not be costly in terms of computation time.

We need, however, to ensure good matching accuracy for the edge areas. Because edges represent the image area with abrupt disparity changes, additional strategies for eliminating false matches must be adopted.

The technique of thresholding is to set a threshold on correlation scores for acceptable potential matches. Since the correlation function ZNCC normalizes the score to be between -1 and 1, we can select a value σ within the range as a threshold. Any candidate match with a correlation score less than the threshold will be discarded. If a pixel from the edge area in the left image has all its candidate pixels in the right image with correlation scores less than σ, then this pixel will be considered unmatched.

It is possible that the match for a pixel p(left image) is a pixel r(from the right image) that has been already matched to another pixel q(from the left image). In this case, we are in a multiple matching situation where a pixel in one image is being matched to two or more pixels in the other image. For instance, both p and q from the left image have r as the one with which the best correlation is generated.

The ambiguity of multiple matching should be eliminated by enforcing the uniqueness constraint. We have adopted the strategy of winner-take-all to solve this problem. If the match r found has been matched earlier to q, we compare the correlation scores of the pairs (p, r) and (q, r). The one with the higher score is the winner and will take r as the match. The loser has to select the pixel that produces the second highest correlation score as its match. This process can be repeated if the loser pixel ends up in another matching conflict with another pixel. However, the repetition of this process

invoked by one multiple matching situation should be terminated if it has repeated for certain times. In this case the current loser will be considered unmatched. Table 5.1 gives the pseudo-code description for the algorithm. Figure 5.5 shows the result of matching edge areas(Figure 5.4).[1]

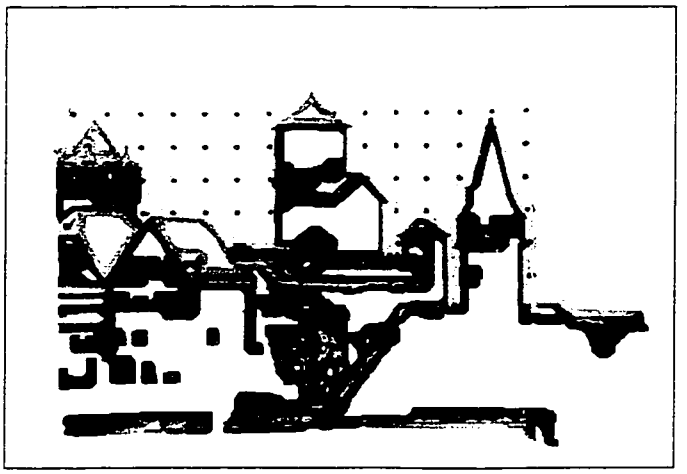


Figure 5.5: Matching result of edge areas.

[1]The matching result is shown by using a reconstructed target image.

```
Match(p)
Begin
For each p ∈ edge area of LeftImage, do
    match(p) = maxZNCC(p,q), q ∈ search area in RightImage;
If (match(p) < σ)
   p = unmatched;
Else
   If ( Verify(p,q) == True)
      p matches q;
   Else
      match(p) = SecondmaxZNCC(p,q), q ∈ search area in RightImage;

End

Verify(p,q)
Begin

If (q matches null)
   return true;
Else    q matches p/;
   If ( ZNCC(p,q) > ZNCC(p/, q) )
      match(p/) = SecondmaxZNCC(p/,q), q ∈ search area in RightImage;
      return true;
   Else
      return false;
End
```

Table 5.1: Pseudo code for eliminating false match in edge area.

5.3 Matching Non-edge Area

The non-edge areas are the remainder of the image after the edge areas are extracted. To match non-edge areas in the left image, we first need to introduce the term of non-edge segment. Since the matching is carried out along epipolar lines, we define a non-edge segment as a sequence of non-edge pixels on an epipolar line, delimited by two edge areas. As shown in Figure 5.6, the white areas represent the edge areas and gray areas the non-edges. Consider the points A, B, C, D, E and G along the epipolar line that crosses the whole image. One can note that we have three non-edge segments; namely the segments AB, CD and EG.

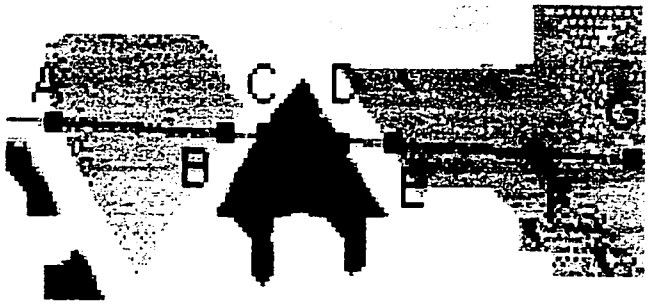


Figure 5.6: Defining non-edge segment.

Since every pixel from the non-edge area falls into a certain non-edge segment, the matching of image's non-edge areas can be transformed into the matching of all the non-edge segments on all the epipolar lines. Obviously the match of the pixels of a non-edge segment must lie on the same corresponding epipolar line in the other image. Furthermore, since the non-edge areas consist mainly of smooth surfaces without abrupt disparity changes, we can enforce all the following constraints: order, continuity and uniqueness. As a consequence, the search area for the matching is drastically reduced. In addition, the likelihood of a mismatch is also reduced.

5.3.1 Enforcing All Constraints

Consider the matching process of a non-edge segment L of the left image to its corresponding pixels in the right image. Let $L[N]$ denote the N pixels of this segment and

let R be the corresponding epipolar line in the right image. If we have a matched pair of pixels such as

$$L[i] \longleftrightarrow R[j]$$

where we refer to $L[i]$ as the *reference point*, its match $R[j]$ the *reference match*, and $L[i]$ $R[j]$ together the *reference pair*.
then

$$L[i+1] \longleftrightarrow R[j+1].$$

The above says that the reference point's right immediate neighbour on the epipolar line matches to the right immediate neighbour of the reference match, while ignoring the scaling that occurs in an image.

As the order constraint states, the order of matching is preserved along the epipolar line, which means the pixel P on the right of the reference point must be matched to a pixel on the same side of the reference match. Moreover, since we are matching the non-edge area where disparity varies smoothly, the disparity of P is almost the same as the disparity of the reference point. Therefore, the match of P should be the neighbour of the reference match on the same side.

However, considering scale change of the image, we need to add a margin δ , so the matching relation becomes:

$$L[i+1] \longleftrightarrow R[j+1+\delta],$$

where $\delta \in 0, 1, ...k$.

This δ represents a small neighbourhood around the reference match. We choose $k = 3$ which means that the reference point's immediate neighbour should fall into the region of less than 3 pixel on the same side of the reference match. This is a threshold that depends on how much the second image has stretched or shrunk with respect to the first one.

As shown in Figure 5.7, pixels i and j represent a reference pair. The possible match of pixel $i+1$ on left epipolar line must be around the right neighbourhood of

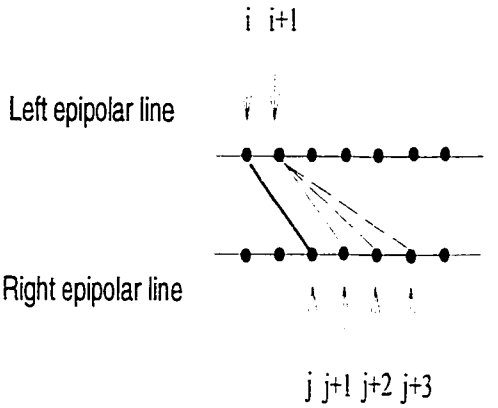


Figure 5.7: Combining constraints to match non-edge area I.

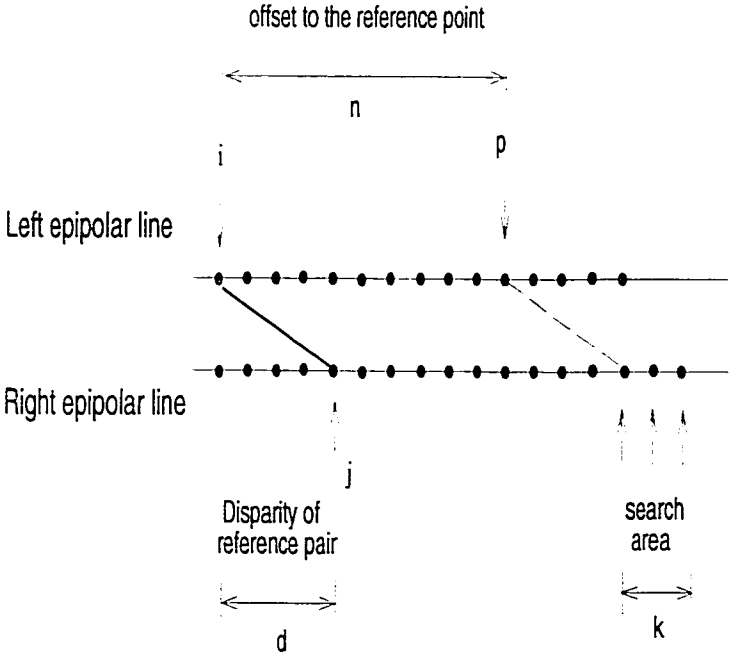


Figure 5.8: Combining constraints to match non-edge area II.

j. Using our threshold $k = 3$, the match of $i + 1$ belongs to the 3-element set $\{j + 1, j + 2, j + 3\}$.

Now, we can apply this method to the n'th neighbour of the reference point within the same non-edge segment as follows:

for a matched pair

$$L[i] \longleftrightarrow R[j],$$

we have

$$L[i + n] \longleftrightarrow R[j + n + \delta],$$

where $L[i + n], L[i] \in$ same non-edge segment.

Since the sharp disparity change only occurs at edges, and according to the definition of the non-edge segment there is no edge point intersecting the same non-edge segment, we can conclude that the disparity within a non-edge segment varies very little. Therefore, the disparity of the reference pair provides an accurate base for matching other pixels in the same segment. To match a certain pixel, the search area can be located by the disparity of reference pair plus its offset from the reference point. The search area for a candidate match is now restricted to a range of very few pixels(three if we set $k = 1$). Within this limited area, we can then simply choose the local maxima using correlation function to select the best match. In Figure 5.8, the search area for point p is obtained by adding the offset n to the reference disparity d. The size of search area is equal to k.

5.3.2 Selecting Reference Pair

The accuracy of the disparity for the reference pair is critical to the matching of the other pixels on the same non-edge segment. Since the search area for a candidate match is limited to a very small number of pixels, the result will be greatly effected by the location of this small area. This location is based on the disparity value of the reference pair. Therefore, we need to adopt reliable strategies to ensure the quality of the reference pair.

A confidence measure and a threshold are used for this purpose. This confidence

```
Match(p)
Begin
In non-edge segment p[N]
count = 0;
For (i=0, i<N; i++)
   match(p[i]) = maxZNCC(p,q), q ∈ search area in RightImage;
   If ( match(p[i]) > σ)
      count++;
   else
      count--;
   if (count> λ)
      p[i] and match(p[i]) are selected as reference pair
End
```

Table 5.2: Pseudo code for select reference pair.

measure is based on the neighbourhood of the candidate reference match. That is, if we can have a certain number of good matches successively, the last one of these matches is selected as a reference pair. The measure of a good match is the threshold value of the correlation score.

The selection of a reference pair is required in two situations. First at the beginning of each non-edge segment because a reference pair should be determined for matching the following other pixels. However, the reference pair needs to be refreshed to avoid accumulated misalignments. The smoothly changing disparity could be accumulated over a certain range and may cause the search area to be drifted away from the correct match. Therefore, over a wide non-edge segment, we do not use the same reference pair. Instead, a new reference pair is established after each time the reference pixel becomes far away from the current pixel to be matched. This distance represents another threshold that we can use to avoid the drifting phenomena. An example is shown in Figure 5.6. The initial reference point for the non-edge segment EG is E. However, because EG is too long, we introduced another reference point F. Although F is not the beginning of the non-edge segment, it is used to refresh the reference disparity.

5.4 Interpolation

Interpolation is a procedure of estimating missing values within an area of known values. In many feature-based stereo matching algorithms, interpolation is used to calculate the disparity of object surfaces. In those cases, complex interpolation algorithms are required to fill the gaps between the sparse set of matched features.

In our approach, a simple interpolation model is applied to refine the dense match results of both edge and non-edge areas. Given an unmatched pixel with most of its neighbours matched, this pixel's disparity is estimated by taking the simple average of its immediate neighbours' disparity values. The interpolation is first carried out separately for edge areas and non-edge areas. Since discontinuity occurs at object boundaries, interpolation between edge and non-edge areas may cause blurring around edges.

To avoid mistakes, a pixel is interpolated only if it fits one of the required patterns. That is, at least two out of the eight neighbours of an unmatched pixel should be matched and in certain positions. These patterns are shown in Figure 5.9, where the grey block represents the pixel to be interpolated and black the already matched pixel. By interpolation, The average disparity value of the neighbouring pixels is assigned to the unmatched pixel in the center.

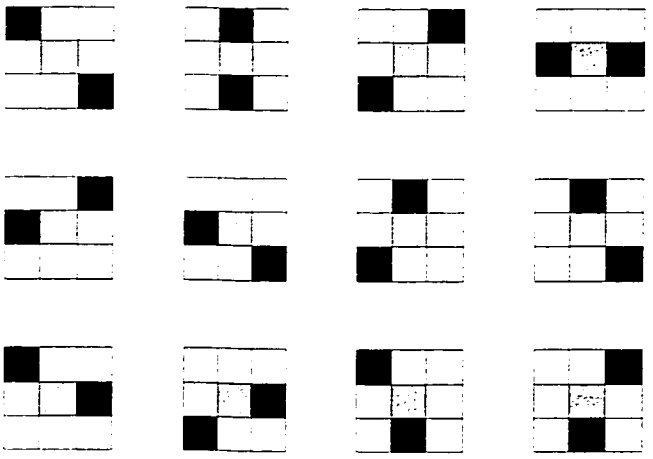


Figure 5.9: Patterns for interpolation.

5.5 Experimental Results

We have tested our methods on four pairs of real images. In our experiments, the edge area includes 3 pixels around the edge output of the Deriche's edge detector. For a better comparison, uninterpolated and interpolated results are both included. The results have shown significant improvements in the matching speed and quality.

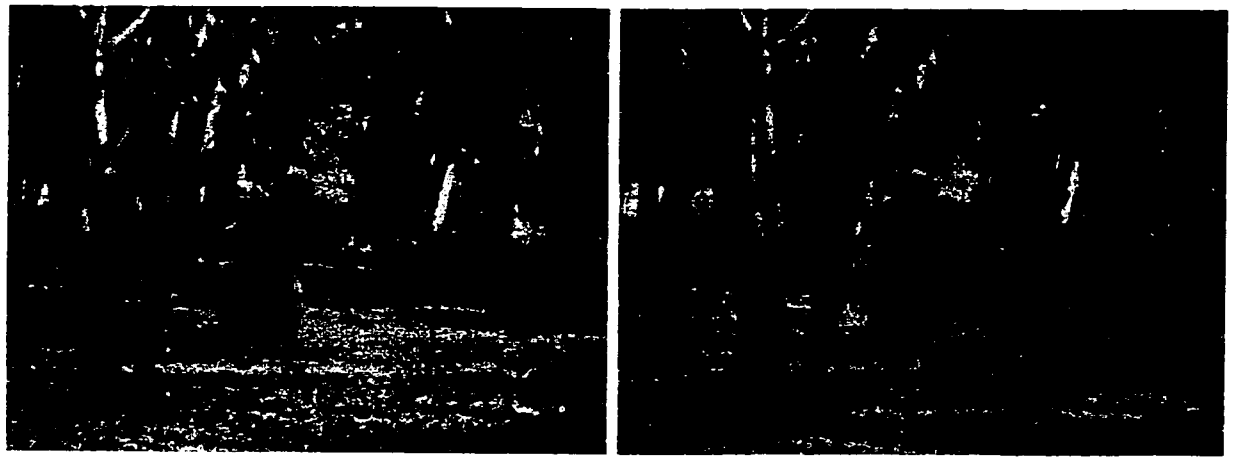
For the Castle scene, we compare the results of using the hybrid method without interpolation(Figure 5.11, 5.12), with the results using interest points(Figure 4.6), with the same size of correlation window. Our hybrid method produced better matching results in less CPU time. Especially, the hybrid method is more effective for recovering object boundaries. Notice the wide white strip at the upper part of the image using hybrid method with bigger window size(11×11, 13×13, 15×15); this has occurred because the pixels on the uniformed background don't produce correlation scores above the threshold setting for acceptable matches, and those pixels are labeled as unmatched.

The Head scene is an indoor scene with objects of more depth differences; there are more edges in the image as well. Compared to the interest point approach, the hybrid method again demonstrates its capability of preserving discontinuity around edges. The number of unmatched points, as well as the overall computing time, is drastically reduced.

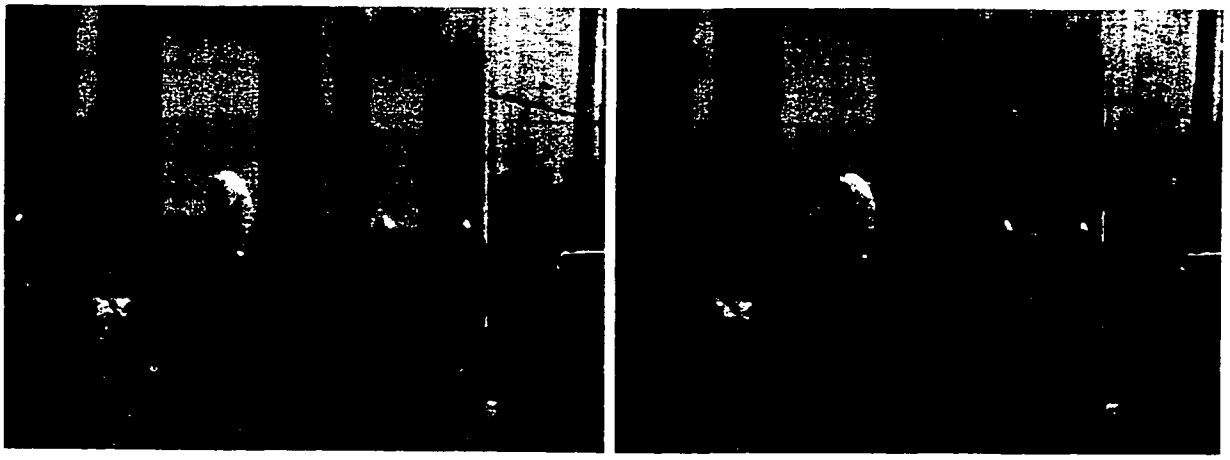
More experiments have been performed on the Tree scene and Meter scene. For the latter, the dense matching process takes less then 10 seconds using a small window(5×5), and the result shown in Figure 5.17 is very impressive.

The comparison of the interest point approach and hybrid approach for the Castle scene and Head scene respectively is illustrated in Figure 5.19 and 5.20. Since there are less edges in the Castle scene, the computing time is reduced more in percentage than for the Head scene.

The CPU time on a Sun Ultra10 workstation of all examples is summarized in Table 5.3. Compared to most early methods which take more than 1 hour to compute a dense matching for images of size 512×512, our approach achieves significant improvement. Moreover, our method can be applied to different real scenes; it is not restricted to stereo images of a certain type.

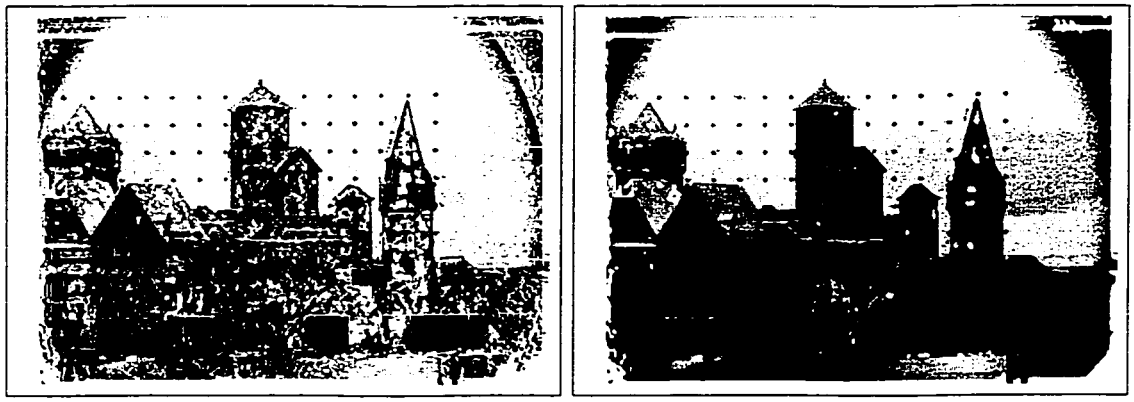


Tree scene

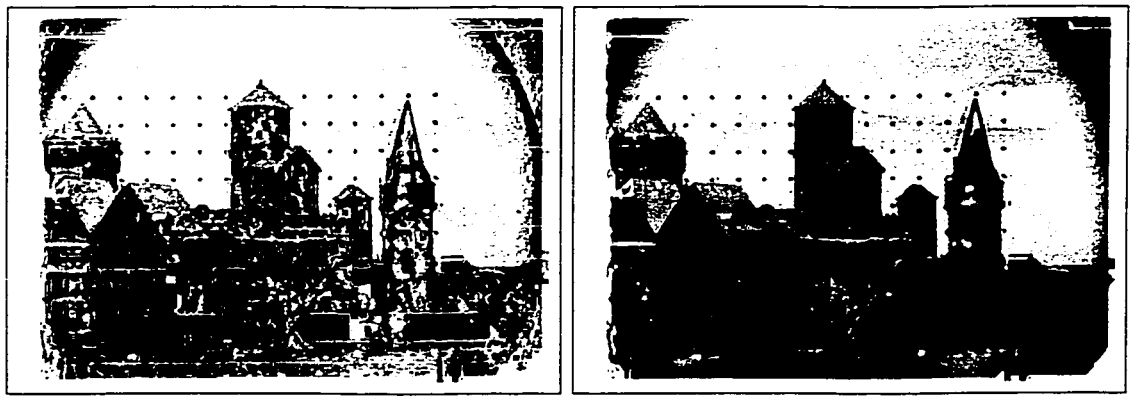


Meter scene

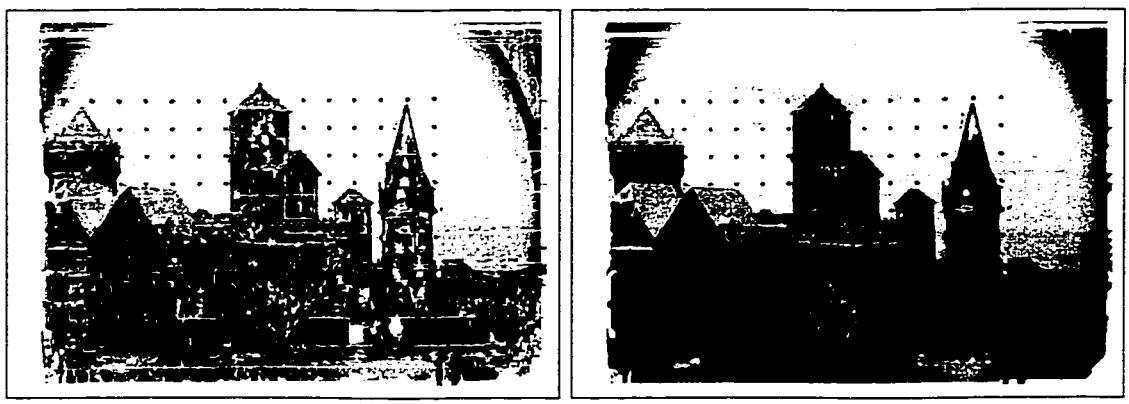
Figure 5.10: Stereo image pairs: Tree scene and Meter scene.



correlation window size 5×5

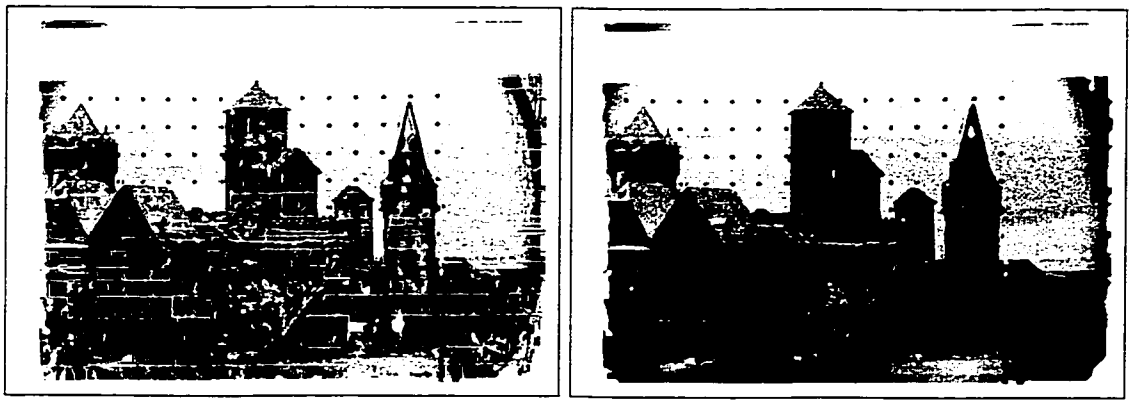


correlation window 7×7

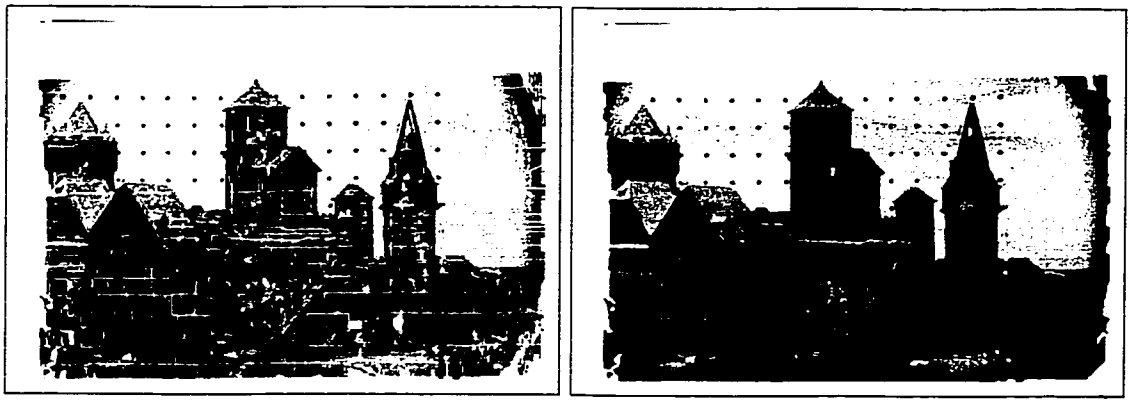


correlation window 9×9

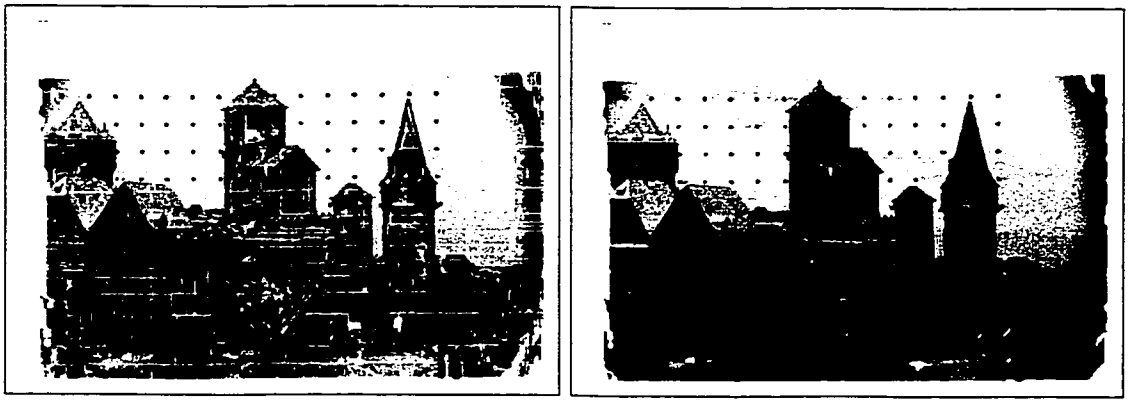
Figure 5.11: Matching results using hybrid approach for Castle scene I; without and with interpolation.



correlation window 11 × 11



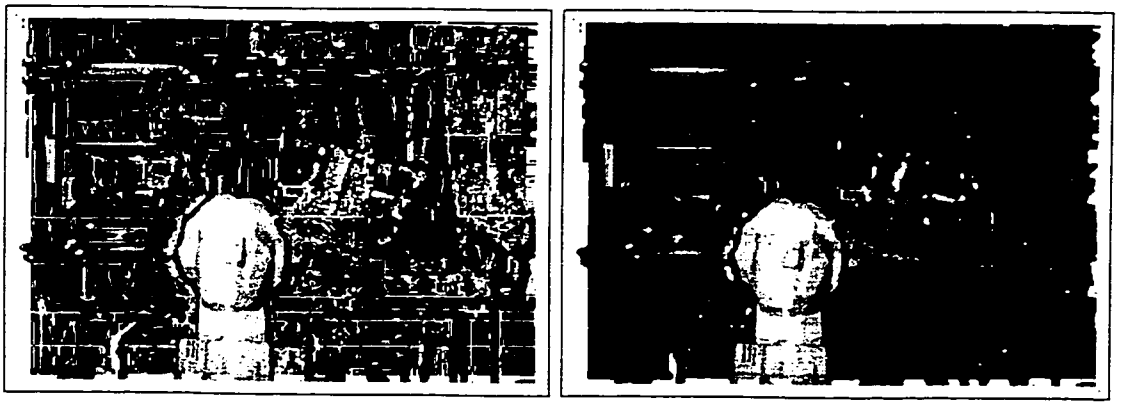
correlation window 13 × 13



correlation window 15 × 15

Figure 5.12: Matching results using hybrid approach for Castle scene II; without and with interpolation.

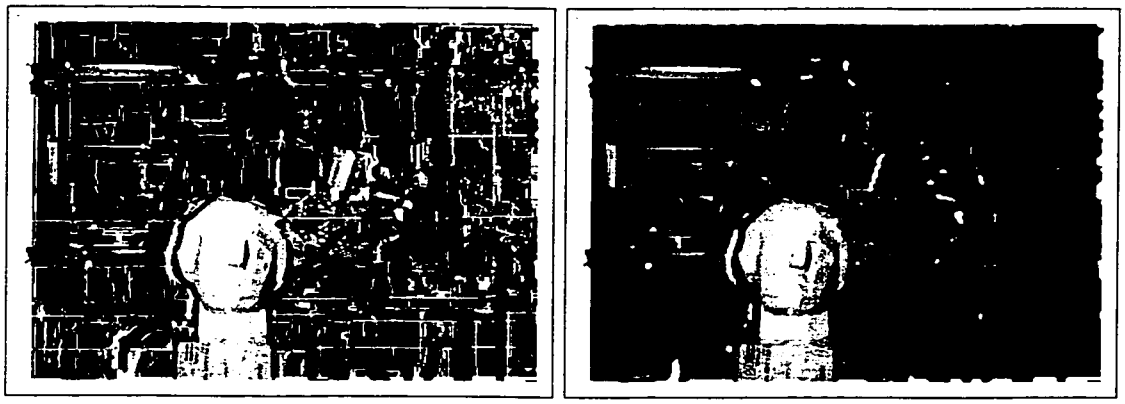
correlation window 5×5



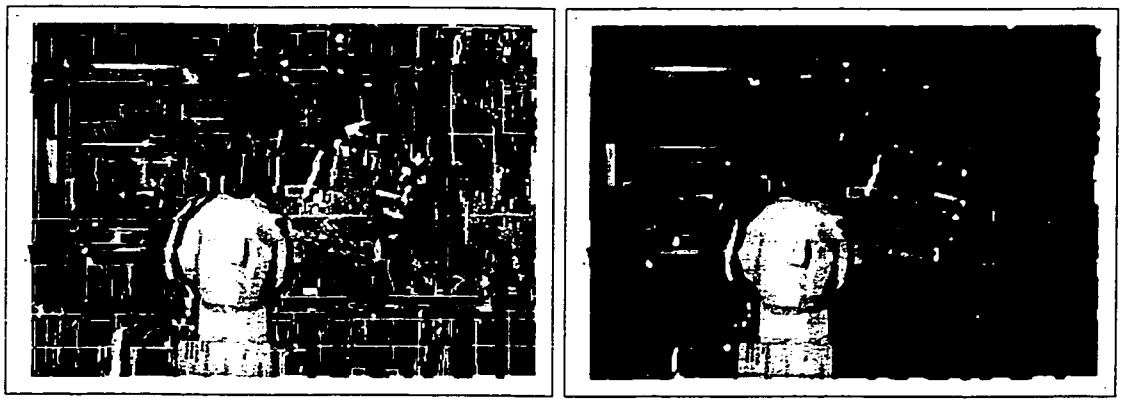
correlation window 7×7

correlation window 9×9

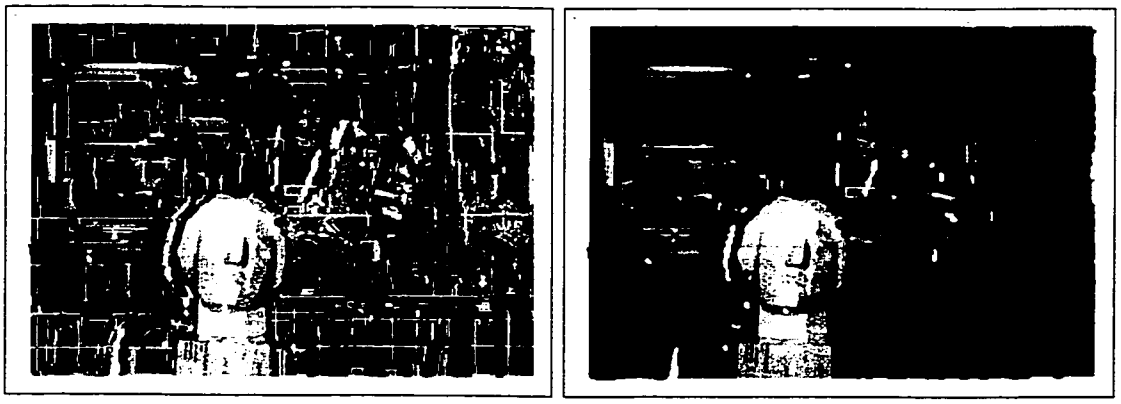
Figure 5.13: Matching results using hybrid approach for Head scene I; without and with interpolation.



correlation window 11 × 11

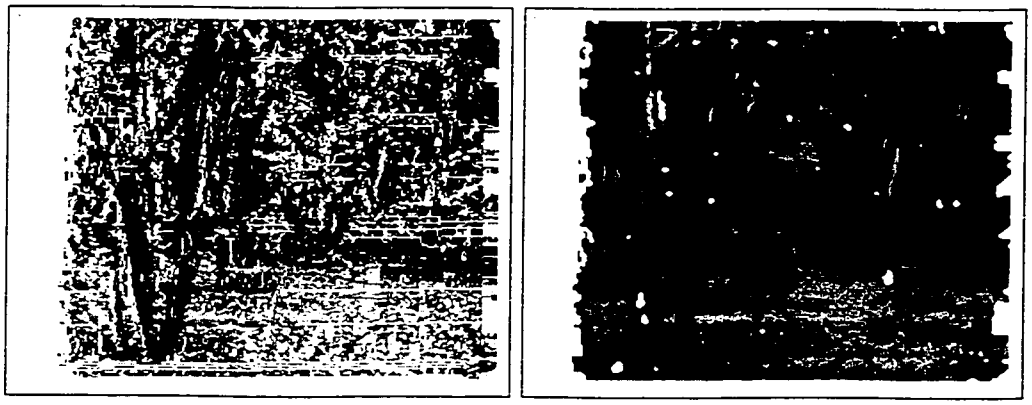


correlation window 13 × 13

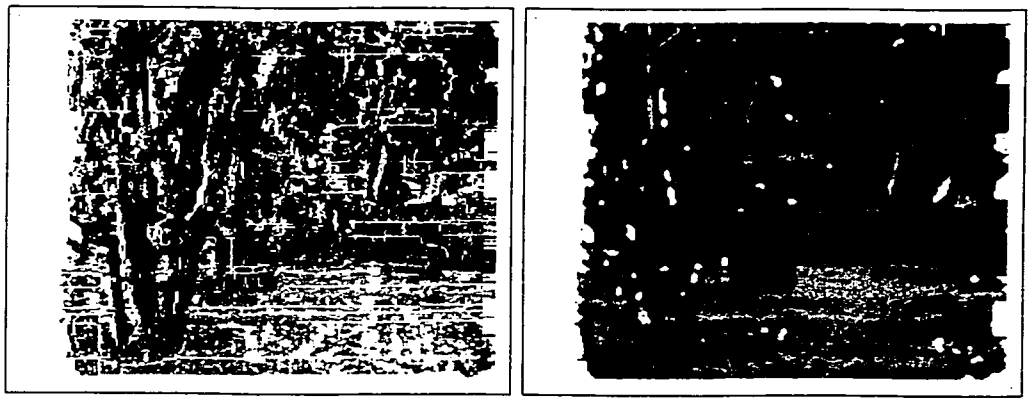


correlation window 15 × 15

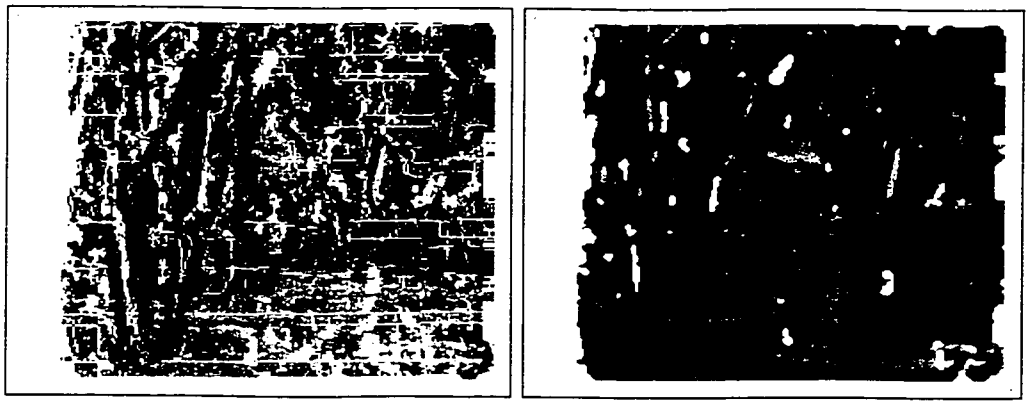
Figure 5.14: Matching results using hybrid approach for Head scene II; without and with interpolation.



correlation window 5×5



correlation window 7×7

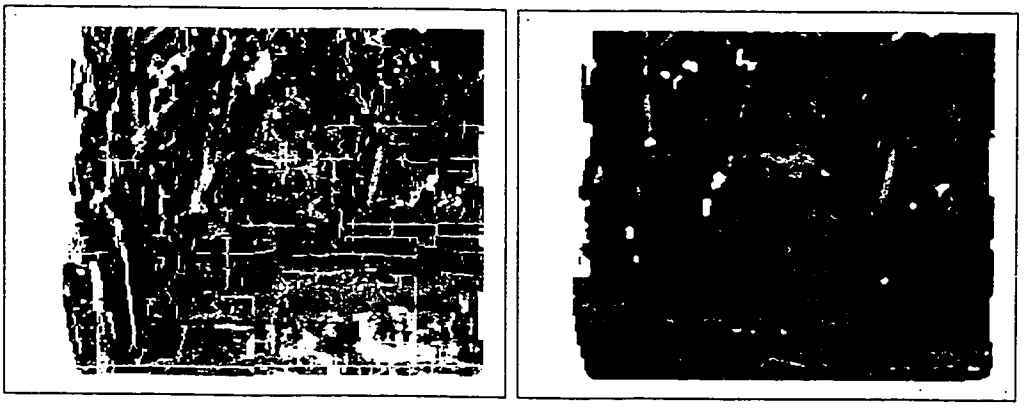


correlation window 9×9

Figure 5.15: Matching results using hybrid approach for Tree scene I; without and with interpolation.

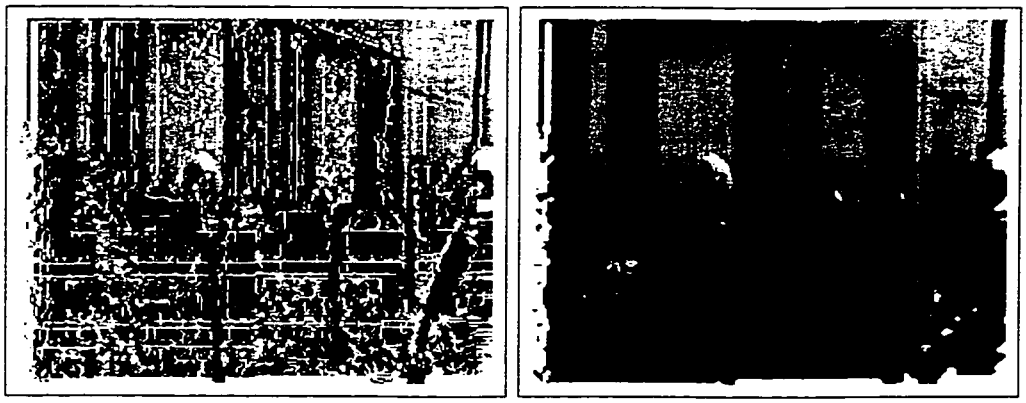
correlation window 11 × 11

correlation window 13 × 13

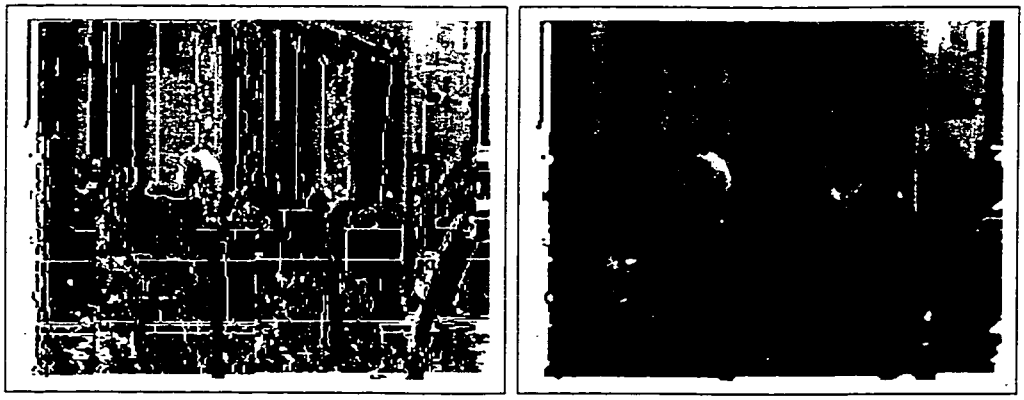


correlation window 15 × 15

Figure 5.16: Matching results using hybrid approach for Tree scene II; without and with interpolation.



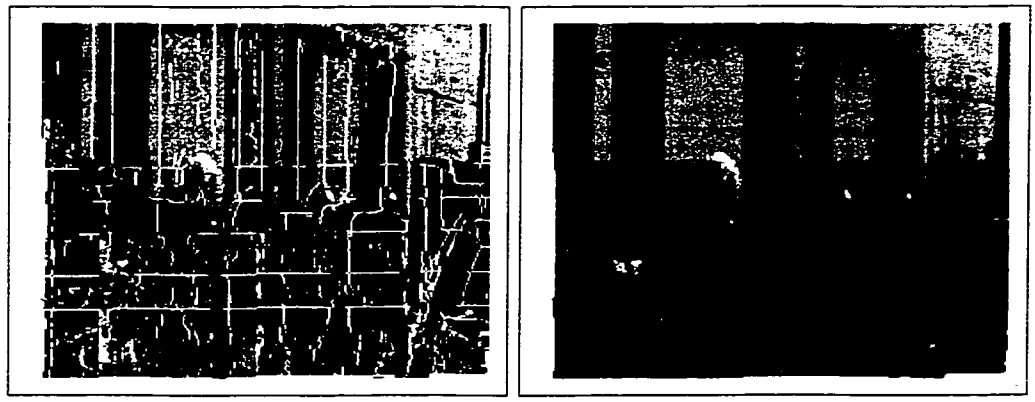
correlation window 5 × 5



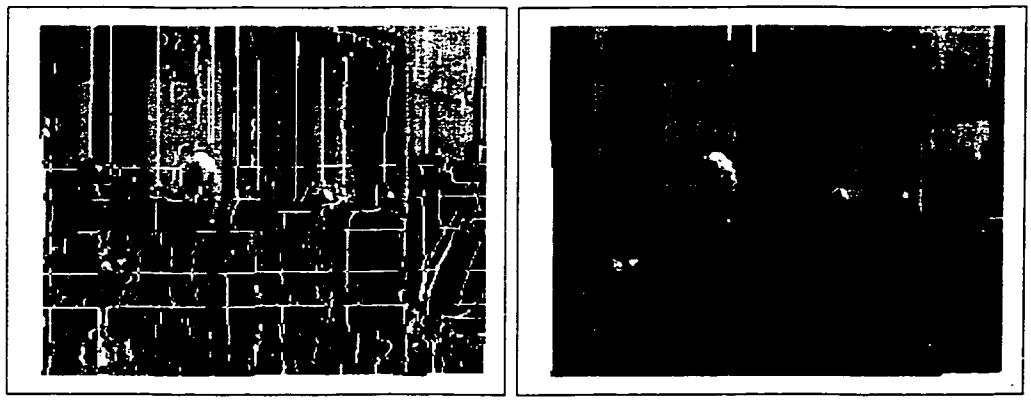
correlation window 7 × 7

correlation window 9 × 9

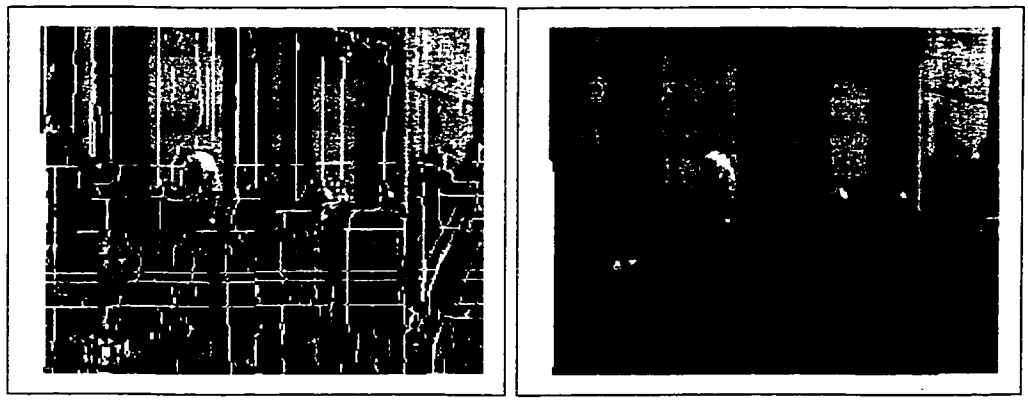
Figure 5.17: Matching results using hybrid approach for Meter scene I; without and with interpolation.



correlation window 11 × 11



correlation window 13 × 13



correlation window 15 × 15

Figure 5.18: Matching results using hybrid approach for Meter scene II; without and with interpolation.

Castle scene (image size 576 × 384)

	correlation window size					
	5 × 5	7 × 7	9 × 9	11 × 11	13 × 13	15 × 15
no interpolation	53s	77s	103s	113s	139s	171s
with interpolation	54s	78s	104s	114s	139s	172s

Head scene (image size 384 × 288)

	correlation window size					
	5 × 5	7 × 7	9 × 9	11 × 11	13 × 13	15 × 15
no interpolation	27s	41s	54s	72s	93s	119s
with interpolation	27s	41s	55s	73s	95s	120s

Tree scene (image size 256 × 233)

	correlation window size					
	5 × 5	7 × 7	9 × 9	11 × 11	13 × 13	15 × 15
no interpolation	12s	21s	32s	46s	61s	77s
with interpolation	12s	21s	32s	47s	62s	78s

Meter scene (image size 256 × 240)

	correlation window size					
	5 × 5	7 × 7	9 × 9	11 × 11	13 × 13	15 × 15
no interpolation	9s	14s	20s	28s	35s	44s
with interpolation	9s	14s	21s	29s	35s	45s

Table 5.3: Processing time using our hybrid approach for all examples.

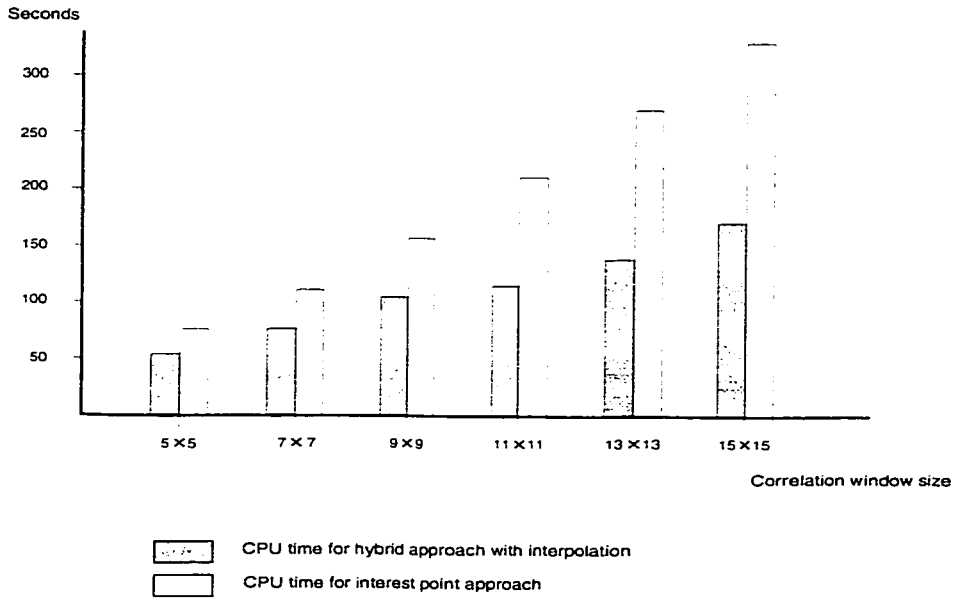


Figure 5.19: Comparing processing time for interest point approach and hybrid approach on Castle scene.

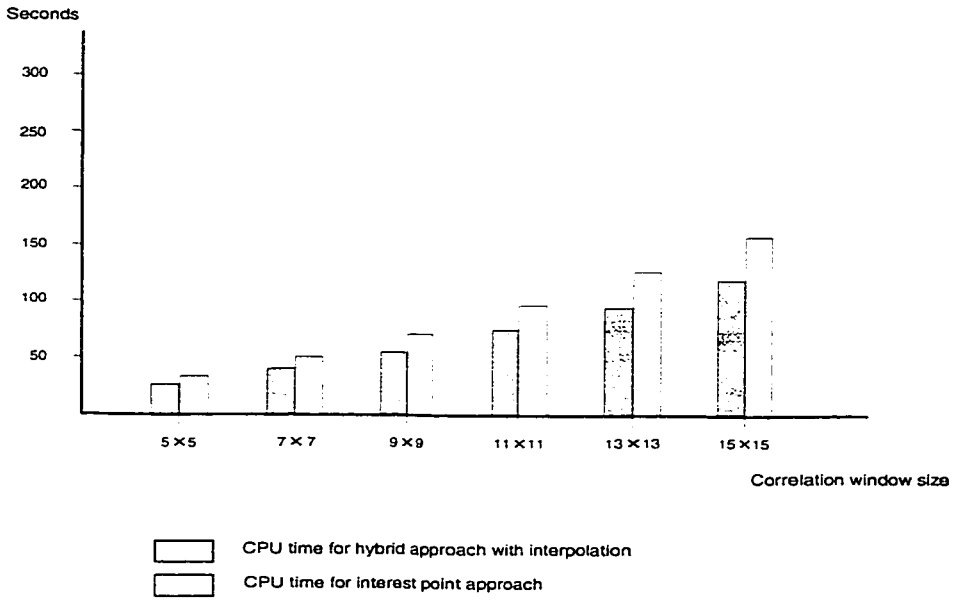


Figure 5.20: Comparing processing time for interest point approach and hybrid approach on Head scene.

5.6 Conclusion

In this chapter, we proposed a fast algorithm for matching uncalibrated images. The algorithm integrates image edge features. Compared to interest points, edges in an image reveal more information of the image structure and indicate the locations of abrupt disparity changes. Since an image is now segmented into two parts: edge areas and non-edge areas, different matching approaches can be applied. The edge areas are matched based on the interest point approach. For the majority of the image which is composed of non-edge areas, we combine all the matching constraints and restrict the search area to only a few pixels. Thus, a more efficient and accurate searching strategy is achieved.

The experimental results demonstrate the capability of our hybrid approach. The overall matching quality is improved: the image object boundaries are matched with better accuracy, and the number of unmatched points is reduced. Meanwhile, the matching speed is considerably increased. Moreover, our hybrid approach can be applied to a wide range of image types with consistent performance.

Chapter 6

Conclusion

The main contribution of this thesis is the design and implementation of a fast dense matching method for uncalibrated images.

Dense matching of uncalibrated images requires the establishment of correspondences between all the pixels of a stereo image pair, when neither the camera's intrinsic parameters nor the camera's position/orientation is available. Using traditional area-based correlation methods to solve this problem is very time consuming and easily influenced by variations between the stereo images. It also suffers from the sensitivity to repetitive patterns. A better way to tackle the matching problem is through the use of hybrid methods, which integrates image features in the matching process. Because image features such as corners, can be detected and matched with higher reliability and accuracy, integrating them in the dense matching provides beacons and safe guards to help guide the whole matching process for every pixel. Hybrid methods introduce the advantages of feature-based matching to area-based matching. Such advantages include lower processing time and robustness against image noise.

In Chapter 4, we used image interest points to provide a disparity estimate. However, using interest points alone is not enough given the complexity of images. In Chapter 5, we designed a hybrid dense matching algorithm for uncalibrated images. This method integrates image edge features, since edges represent the structure of an image and the locations of disparity discontinuity. The matching process is carried

out separately on edge areas and non-edge areas of the image. For matching non-edge areas, we have applied all geometrical matching constraints, including the continuity constraint, to reduce the search range to a minimum size. The overall computational time is drastically reduced.

Our hybrid method has been tested on both indoor and outdoor real scenes. Experimental results have demonstrated its capability and efficiency. We believe that our method is faster and more practical than those reported in the literature. A lot of fast dense matching methods are tested on special types of images and their performance tends to degrade when general images are used. Our method, on the other hand, does not assume any limitation on the observed scene. It applies consistently to scenes with a lot of depth changes as well as to scenes that are almost flat.

We should note, however, that because our method relies on the epipolar geometry that is calculated based on interest points, its accuracy might affect the outcome of the dense matching process. Although new methods for calculating the fundamental matrix F are very reliable, there is no guarantee that F is accurate each time.

Future work

This work might be improved and extended; in particular we see the following possible improvements and extensions:

- The matching of edges, although fast, can be made even faster by using a different edge matching strategy. We have used the traditional area-based correlation approach to match edges; however, the processing time might be reduced by adding constraints on the search areas. In particular, one should investigate the possibility of limiting the search to the edges in the other image.

- The result of our dense matching can be used for the detection of occlusions in images. This is an important application that can be used to identify different objects in a scene.

References

[1] L. Alvarez, R. Deriche, J. Sanchez, and J. Weickert. Dense disparity map estimation respecting image discontinuities: A PDE and scale-space based approach. *INRIA Report*, (3874), 2000.

[2] P. Aschwanden and W. Guggenbuhl. Experimental results from a comparative study on correlation-type registration algorithms. *Robust Computer Vision*, pages 268–282, 1992.

[3] S. T. Barnard and M. A. Fischler. Computational stereo. *Computing Surveys*, 14(4):553–572, 1982.

[4] S. T. Barnard and W. B. Thompson. Disparity analysis of images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2(4):333–340, 1980.

[5] B. Boufama and R. Mohr. A stable and accurate algorithm for computing epipolar geometry. *International Journal of Pattern Recognition and Artificial Intelligence*, 12(6):817–840, 1998.

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, 1999.

[7] J. F. Canny. A computational approach to edge detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(11):769–798, 1986.

[8] S. D. Cochran and G. Medioni. 3D surface description from binocular stereo. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 14(10):981–994, 1992.

[9] R. Deriche. Using Canny's criteria to derive an optimal edge detector recursively implemented. *International Journal of Computer Vision*, 2(4):167–187, 1986.

[10] K. Do, Y. Kim, T. Uam, and Y. Ha. Iterative relaxational stereo matching based on adaptive support between disparities. *Pattern Recognition*, 31(8):1049–1059, 1998.

[11] P. Fua. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sydney, Australia*, August 1991.

[12] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[13] W. E. Grimson. From images to surfaces: A computational study of the human early visual system. *M.I.T.Press*, 1981.

[14] H. J. Hannah. Sri's baseline stereo system. *Proceedings of the DARPA Image Understanding Workshop*, pages 149–155, 1985.

[15] R. Hartley. In defence of the eight-point algorithm. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[16] D. Hearn and M. P. Baker. *Computer graphics.* Prentice Hall, 1986.

[17] W. Hoff and N. Ahuja. Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 11(2), 1989.

[18] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, 1994.

[19] E. Krotkov, M. Hevert, and R. Simmons. Stereo perception and dead reckoning for a prototype lunar rover. *Autonomous Robots*, 2(4):313–331, 1985.

[20] K. S. Kumar and U. B. Desai. New algorithms for 3D surface description from binocular stereo using integration. *Journal of the Franklin Institute*, 331B(5):531–554, 1984.

[21] M. D. Levine, D. A. O'Handley, and G. M. Yagi. Computer determination of depth maps. *Computer Graphics and Image Porcessing*, 2(4):131–150, 1973.

[22] S. A. Lloyd. Stereo matching using intra- and inter-row dynamic programming. *Pattern Recognition Letters*, 4:273–277, 1985.

[23] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 1976.

[24] D. Marr and T. Poggio. A theory of human stereo vision. *A.I Memo*, (451), 1977.

[25] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2):139–154, 1985.

[26] H. Saito and M. Mori. Application of genetic algorithms to stereo matching of images. *Pattern Recognition Letters*, 16(1):815–821, 1995.

[27] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[28] D. Terzopoulos. Regulation of inverse visual problems involving discontinuities. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(4):413–424, 1986.

[29] E. Trucco and A. Verri. *Introductory Techniques for 3D Computer Vision*. Prentice Hall, Edinburgh,UK, 1998.

[30] J. Weng, N. Ahuja, and T. S. Huang. Matching two perspective views. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 14(8):806–825, 1992.

[31] N. Yokoya. Dense matching of two views with large displacement. In *Proceedings of the 1st* IEEE *International Conference on Image Processing*, pages 213–217, 1994.

[32] A. L. Yuille and T. Poggio. A generalized ordering constraint for stereo correspondence. *A.I Memo*, (777), 1984.

[33] Z. Zhang. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.

Vita Auctoris

NAME: Hongxuan JIN

EDUCATION: Shanghai University, Shanghai, China
1993-1997 B.Eng.

University of Windsor, Windsor, ON, Canada
1999-2001 M.Sc.