SHAPE FROM ROTATION USING STEREO

by
Yizhi Edith Li

A thesis submitted
in conformity with the requirements
for the degree of Master of Science

Department of Computer Science
York University
April, 1992

Abstract

This thesis examines the construction of a detailed 3D surface model from an object rotating in front of a stationary camera. An algorithm is developed which integrates repeated stereo views of a rotating object into a 3D model. Starting with disparity estimates obtained using a stereo algorithm, the algorithm estimates the depth of the recovered point, and then projects individual measurements into 3D points. These points are then used to construct an octree representation of the object. The resulting representation provides a full 3D representation of the objects visible surfaces.

Acknowledgments

I would like to thank Professor Michael Jenkin for his supervision of my thesis. Many helpful suggestions, and comments he provided through countless discussions have been valuable asset, for which I am truly grateful. I would also like to thank Bernard Kwok for helping me to get the graphical display package working properly.

Thanks also goes to Professor Minas Spetsakis, Professor Evengelos Milios, Professor Michael McNamee and Professor George Tourlakis for their useful comments and suggestions on my thesis draft.

Help from the Computer Science technical team and Patricia Plummer is greatly appreciated.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

> "What does it mean, to see? The plain man's answer (and Aristotle's, too) would be, to know what is where by looking. In other words, vision is the *process* of discovering from images what is present in the world, and where it is. Vision is therefore, first and foremost, an information-processing task, but we cannot think of it just as a process. For if we are capable of knowing what is where in the world, our brains must somehow be capable of *representing* this information—in all its profusion of color and form, beauty, motion, and detail"[1].

Vision is the means whereby we perceive space as a whole. Such a statement does not imply that all space is perceived at once, but rather that it is only by vision that spatial relations of objects outside the body and at a distance from it are satisfactorily dealt with in detail.

The purpose of computer vision is to enable a computer to understand its environment from visual information. Visual information processing holds an important position in the intelligent activities of human beings. Computer vision, therefore, has been of interest as a research theme in artificial intelligence (AI), and has been studied from the early stages of AI research.

Researchers have categorized the general problem of understanding visual information into two main areas: Low-level vision, which deals with the task of extracting primitive measurements from an image, and High-level vision, which deals with the task of inferring complex structures such as particular objects from more primitive visual measurements.

3D computer vision processes consist of many different components, each of which perform particular functions in coordination with other components. First, visual infor-

[1]Marr [22], pp3.

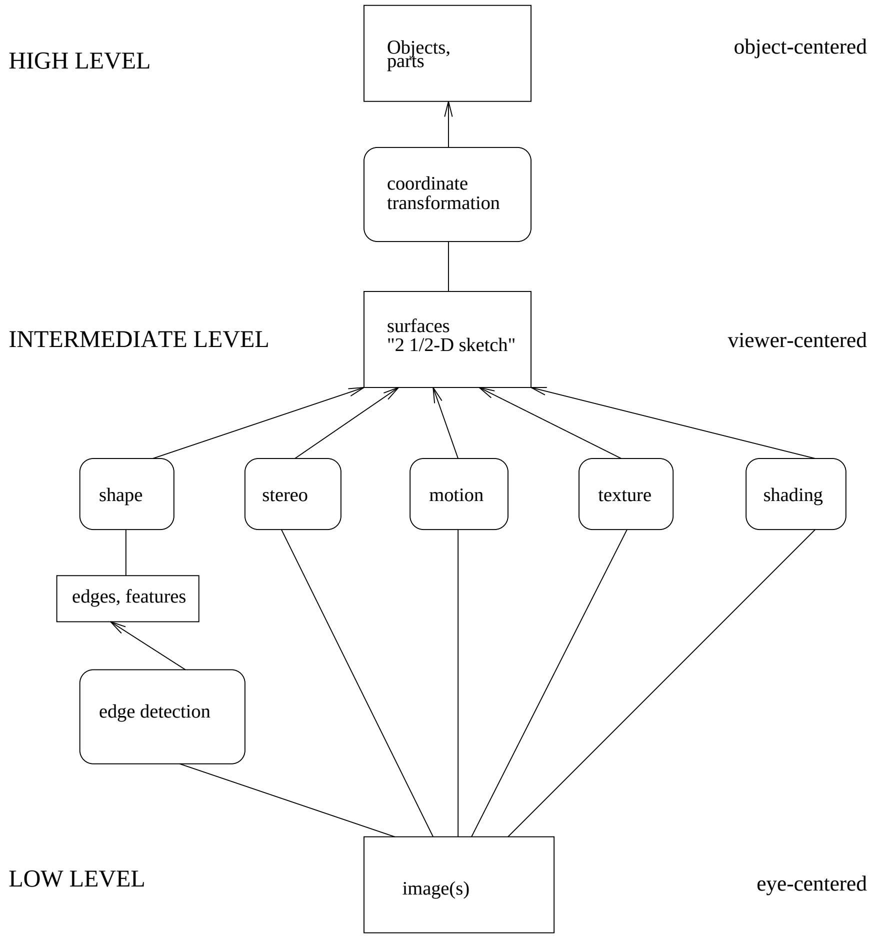

Figure 1.1: Visual processing hierarchy (from [33])

mation must be taken from a 3D scene, such as digitized images. Then these images are analyzed in order to obtain a collection of low level features, such as edges, lines, and regions. These features may be grouped together to form higher level constructs which in turn may be used as the input symbols to object recognition processes.

Low-level visual processing is often characterized as the extraction of *intrinsic images* [3] such as depth, orientation or reflectance from the input intensity images (see Figure 1.1, adapted from [33]). A characteristic of these images is that they often present a *dense field*, i.e., the information is available at all points in the two-dimensional visual field. This dense, visual information is then segmented and grouped into coherent surfaces, parts and objects by later stages of processing.

Intrinsic images form a useful intermediate representation and facilitate the task of higher level visual processing. Intrinsic characteristics such as depth or reflectance are more useful than raw intensities for scene understanding or object recognition since they are closer to the true physical characteristic of the scene. These intermediate representations also provide a framework for the integration of information from multiple low-level vision modules such as stereo, shading, occluding contours, motion, and texture, and for integrating information over time [33].

One of the most important measurements that can be made by a low level visual process is the determination of the depth of a given point in the scene. Without depth information, a large number of object recognition and reconstruction processes become difficult or even impossible. For example, there are an infinite number of possible 3D structures that give rise to the same two dimensional projection when there is no depth information available.

There are many techniques for obtaining the depth information of a projected point in a scene. One of these is the use of stereo vision. If the same object is seen from two different positions, its 3D position can be measured. This technique is called *stereo vision* or *binocular vision*. It is one of the most important ways in which humans capture short range distance information.

Stereo may be applied to computer aided design and manufacturing (CAD/CAM), which are rapidly growing fields in modern industry. The flexibility and speed at which designs can be altered, examined and manufactured make these technologies very popular. One drawback of these techniques is the need to produce computer representations of existing objects in order to be able to integrate new designs with existing equipment. In this thesis, an algorithm is presented which constructs a 3D surface models of an object as the object rotates in front of a stationary video camera. The algorithm processes the images in pairs, and thus the computational task can be thought of as a generalization of convergent stereopsis, in which many views (rather than just two) of the same scene are integrated together in order to obtain a 3D description of the object. Starting with disparity estimates obtained using an existing stereo algorithm, the algorithm estimates the depth of the recovered point, and then projects individual measurements into 3D points. These points are then used to construct an octree representation of the object. The resulting representation provides a full 3D representation of the visible surface of the object. A graphical display package has been developed to manipulate the resulting representation.

The stereopsis algorithm used here provides a relatively dense set of image disparity measurements for each pair of rotated views of the scene without special light sources. Furthermore, using an octree representation in the algorithm allows us to represent and manipulate arbitrarily shaped 3D models in our shape from rotation experiments and to simplify the task of rotating the representation of the object.

The algorithm presented here is an initial step towards the automatic construction of a full 3D surface model of an object, suitable for inclusion within a CAD/CAM model of the environment.

1.2 Outline of the Thesis

Chapter 2 examines some previous work on 3D shape modeling. Chapter 3 presents an overview of some previous work on stereopsis theories and algorithms. Chapter 4 describes in detail the extraction of shape from rotation and its implementation. Chapter 5 shows and analyses the results of the implementation of our algorithm. The thesis is summarized and future research directions are proposed in Chapter 6. A glossary of some mathematical terms and notation used in this thesis is provided in the appendix.

Chapter 2

Previous Work: Object Representation

This chapter provides a brief survey of the existing literature in the area of object representation. The next chapter provides a brief overview of the state of the art in computer algorithms for stereopsis. Existing results in these two fields provide the basis for this thesis.

One of the fundamental research issues in computer vision is 3D shape modeling. The underlying problem is to design or develop representations that are sophisticated enough to model interesting objects, yet which are simple enough to permit recognition or construction from images or other sense data. A variety of surface-based models have been proposed, including generalized cylinders [4], superquadrics [29], deformable finite-element models [39], and volumetric models such as octrees [15,27]. Once a particular representation has been chosen, there still remains the task of extracting visual information in order to build up a description of the object. One popular approach to constructing a representation from a rotating object has been to intersect multiple silhouettes of the object as seen from different views [7,33]. This is a very limited technique as only boundary information is available, thus discarding potential information from the surfaces of the object.

2.1 The ACRONYM Model-Based Vision System

Model-based vision has many applications including in photointerpretation and manufacturing. For example:

1. A photointerpreter programs the system to monitor aircraft at an airport. Aircraft models are entered along with a model of the airport. The system is told that aircraft are found at airports on runways, taxiways, or storage areas.

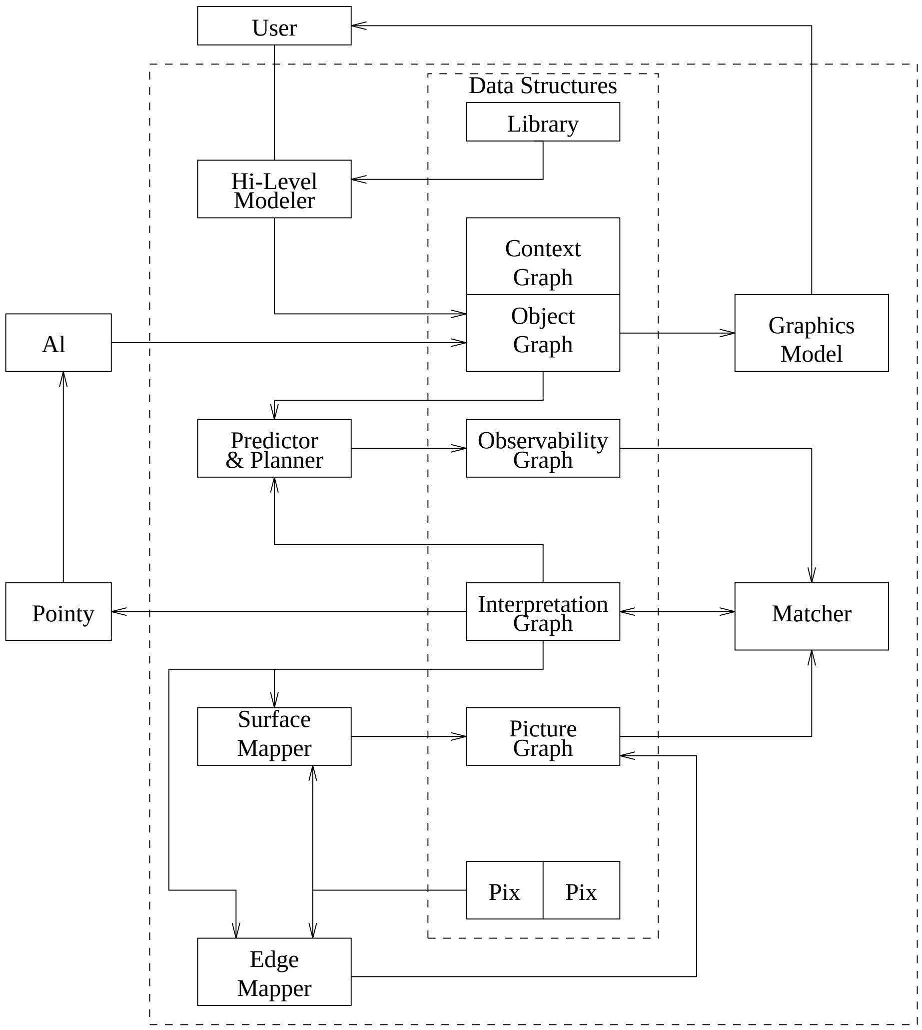

Figure 2.1: The ACRONYM system, adapted from [4]

2. An engineer programs the system to inspect a part, or to pick the part from a bin of identical parts and place it in an assembly. He uses the part model from a CAD system and specifies a model of the workstation.

ACRONYM [4] is a model-based image understanding system. It demonstrates mechanisms for the interpretation of images with generic object classes and generic viewing conditions in a way that is generalizable. It combines a geometric modeling capability with a high level modeling language for natural communication with the user in terms of object models. A user gives high level descriptions of both generic and specific instances of objects. A rule-based inference system produces a viewpoint dependent symbolic summary of the predicted appearance of the objects.

The ACRONYM system (see Figure 2.1, adapted from [4]) can be described as follows: The user interacts with a high-level model to produce both generic and specific models. The result of this modeling process is called the Object Graph. This graph drives a graphics module which provides the user with visual feedback of shape and spatial relations during the modeling session. The Predictor and Planner module produces a symbolic summary of the way in which an object is modeled in the Object Graph and how it can be expected to appear in an image. This summary is called the Observability Graph and is used to program the Matcher from Figure 2.1. The Matcher constructs an interpretation Graph from the Observability Graph and the Picture Graph, which is a symbolic description computed from the image by the surface mapper, and the edge mapper.

The Object Graph of ACRONYM system contains three dimensional models. It provides volumetric representations of objects, both specific and generic, and three dimensional spatial relations between volume elements. The basic volume primitive used is the generalized cone. Spatial relationships of volume elements within an object are defined hierarchically.

A generalized cone is defined by generalized translational invariance. It is described by three sub-primitives: a spine, a cross section and a sweeping rule. In its most general form, a generalized cone is the volume swept out by the cross section as it is translated along the spine, while being deformed according to the sweeping rule.

Unfortunately, such a representation is only capable of an extremely abstract description of most natural and biological forms, as is illustrated in Figure 2.2. It cannot accurately and succinctly describe most natural animate forms or produce a succinct description of complex inanimate forms such as clouds.

2.2 Superquadrics parts models

Pentland [29] has presented a representational system that has been proven competent to accurately describe an extensive variety of natural forms (e.g., mountains, trees, clouds),

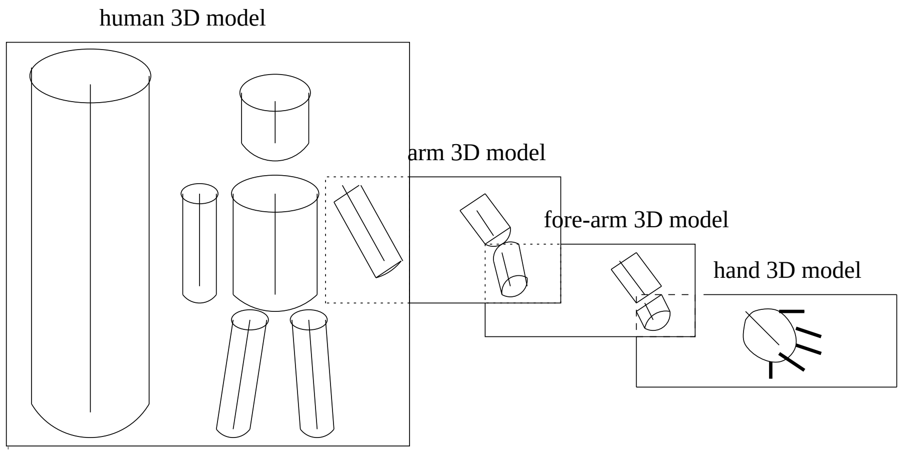

Figure 2.2: Description of human form

as well as man-made forms, in a succinct and natural manner. The idea behind this representational system is to provide a vocabulary of models and operations that will allow us to model our world as the relatively simple composition of component "parts", parts that are reliably recognizable from image data.

In this system, a modeling primitive may be deformed and shaped, but is intended to correspond roughly to our naive perceptual notion of "a part". For this basic modeling element, a parameterized family of shapes known as a *superquadrics* is used. A superquadric is a three-dimensional vector that sweeps out a surface parameterized in latitude and longitude with the surface's shape controlled by two different parameters. This family of functions includes cubes, cylinders, spheres, diamonds and pyramidal shapes as well as the round-edged shapes intermediate between these standard shapes.

These basic "lumps of clay" (with various symmetries and profiles) are used as prototypes which can be deformed by stretching, bending, twisting or tapering, and then combined using Boolean operations ("and", "or", and "not") to form new, complex prototypes that may, recursively, again be subjected to deformation and Boolean combination.

The description language is designed to describe shapes in a manner that corresponds to a possible formative history. For example, how one would create a given shape by combining lumps of clay. Thus the description provides us with an explanation of the image data in terms of the interaction of generic formative processes. This primitive explanation can then be refined by application of specific world knowledge and context. By using such a process-oriented possible-history representation we force the resulting descriptions to group points that have similar causal histories, thus obtaining "parts" that interact with the world in a relatively simple, holistic manner. This further simplifies many reasoning tasks, because the parameters and components that affect interactions tend to be explicitly represented rather than being some complex or difficult-to-calculate function of the description's variables.

This language allows twisting deformations, stretching, bending, and tapering to construct an extremely wide variety of objects. Furthermore, this language allows (hierarchical) Boolean combination of these primitives. One can build a specific object by applying the logical set operations "and", "or", and "not" to component *parts*.

The major limitation of the models are that they are capable of accurately representing only a restricted class of objects because they "wire into the parameterization" a relatively small family of shapes, rather than place generic constraints. The models are not sophisticated enough to deal with all of the complexities of the real world. Furthermore, the models cannot immediately accommodate most objects of interest, so objects must be hierarchically subdivided into compatible primitives that may then need to undergo shape transformations.

This approach for representing the three-dimensional world becomes awkward when applied to complex natural surfaces such as mountains or clouds. The most pronounced difficulty is that the superquadric lumps-of-clay representation becomes implausably complex when confronted with the problem of representing, e.g., a mountain, crumpled newspaper, or a bush. As Pentland noted in [29]:

> "This makes the technique ill-suited to solving the problem of representing *classes* of such objects, or determining that a particular object is a member of that class"[1].

2.3 Deformable finite-element models

Terzopoulos, Witkin and Kass [39] proposed models of 3D shape which may be viewed as deformable bodies composed of simulated elastic material. They developed a deformable model for 3D shape which has a preference for axial symmetry. Symmetry was represented even though the model does not belong to a parametric shape family such as generalized cylinders. A symmetry-seeking property was designed into the internal forces that constrain the deformations of the model. They developed a framework for 3D object reconstruction based on symmetry-seeking models. Examples of these models were formed from monocular image data through the external forces which were derived from the data.

By coupling deformable tubes to deformable spines, models whose deformations exhibited an affinity for certain 3D symmetries were created. The shape constraints intrinsic to the model are formalized by strain energy functionals which dictate the elastic properties of the material [38] and by internal coupling forces which produce the desired symmetries.

[1] Pentland [29], pp308.

The framework for object reconstruction is as follows: Model instances are formed from visual data by combining the internal shape constraints with whatever external constraints may be derived from the data. The internal constraints of the models are generic. The external constraints are interpreted as external forces applied to the models. The framework is based on the dynamics of elastic bodies and reconstruction algorithms can be obtained in a principled way by applying standard numerical solution techniques.

The symmetry-seeking model is described as follows: Imagine a deformable sheet made of elastic material. Take this sheet and roll it to form a tube. Next, pass through the tube a deformable wire spine made of a similar material. At regularly spaced points along the spine, couple it to the tube with radially projecting spring forces to maintain the spine in more or less axial position within the tube. Include additional forces that force the tube into a quasi-symmetric shape around the spine. Finally, provide extra control over shape by introducing expansion/compression forces radiating from the spine. By adjusting the control variables designed into the model, a variety of interesting behavior (including visco-elasticity) can be obtained.

Furthermore, the forces applied to the symmetry-seeking model were derived from monocular images. The applied forces deform a model in 3D space so that its projection into the image plane is consistent with the 2D silhouette of an object of interest. The model reconstructs 3D shapes by achieving a stable equilibrium between the externally applied forces and the internal stresses due to the model's deformation constraints.

The advantages of the models are as follows:

1. Superquadrics parts models are purely geometric, hence *passive*. In contrast, these symmetry-seeking deformable models are not purely geometric, they are *active*. They react to externally applied forces as one would expect real elastic to react because deformable models are governed by variational principles of elasticity theory.

2. Their ability to represent symmetry without sacrificing the possibility of representing detailed shape information.

3. The geometric coverage of deformable models can be significantly broader than the superquadric models developed by Barr [2] and Pentland [29].

However, this model only deals with deformable models for 3D shapes which have a preference for axial symmetry. Its geometric coverage is still quite restricted. Furthermore, its implementation is rather complex.

2.4 Indexing process matching models

Dickinson, Pentland and Rosenfeld [8] have proposed a model for the recovery and recognition of 3D objects from a single 2D image. Their approach is outlined in Figure. 2.3 (adapted from [8]). The model proceeds as follows:

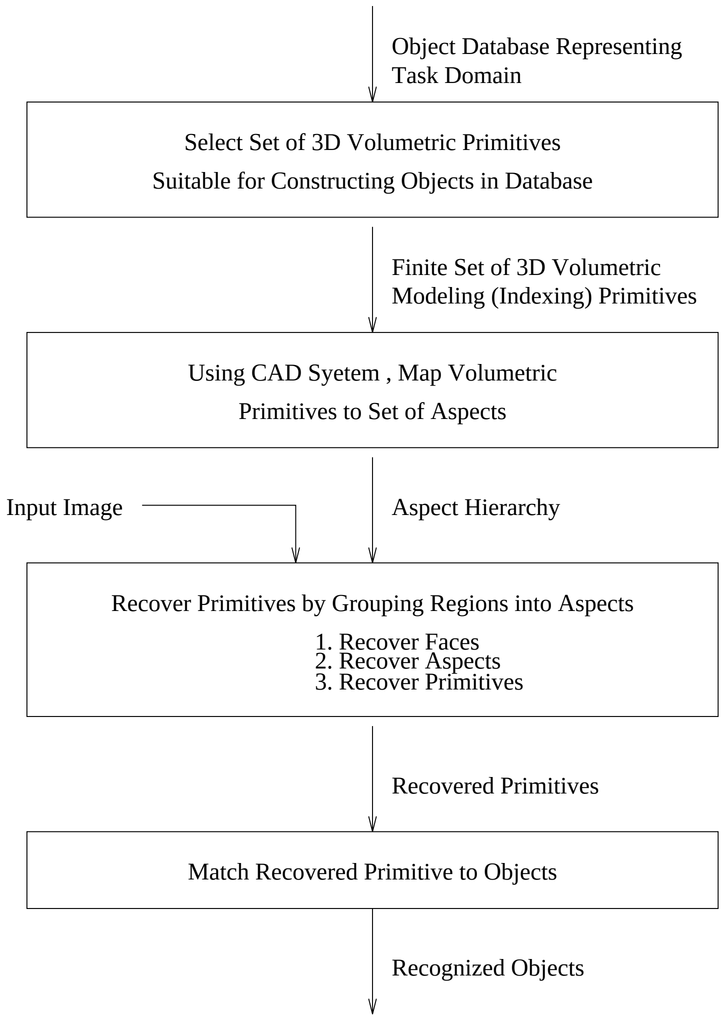

Figure 2.3: Overview of Dickinson's approach, adapted from [8].

Given a recognition task domain consisting of a database of objects, the first step is to select a set of object-centered 3D volumetric primitives which when assembled together can be used to construct the objects. Next, using a CAD system, the set of volumetric primitives are mapped to a set of viewer-centered 2D aspects whose number is fixed and *independent* of the size of the object database. The aspects are represented by a hierarchy of 2D features, called the *aspect hierarchy*, whose levels include the qualitative shapes of the primitives' projected surfaces (faces), subsets of contours that bind the faces (boundary groups), and groups of faces (aspects). The relations between these features are then assessed from all viewpoints, resulting in a table of estimated conditional probabilities for each 2D feature and primitive as a function of less complex 2D features.

Primitive recovery consists of three steps.

1. From a region segmentation of the image, each region is classified (or labeled) according to the faces in the aspect hierarchy; if a face is occluded, it may receive multiple labels.

2. The regions are partitioned into groups, each corresponding to an aspect of a primitive. The problem is formulated as a region labeling, and the conditional probabilities within the hierarchy are used to avoid a combinatorial explosion.

3. The last step consists of mapping the 2D aspects to the 3D primitives, and recovering the primitive connections.

From a set of recovered 3D volumetric primitives, the task of recognizing the objects contained in the scene is then proceeded. Here, groups of recovered primitives, including their connections, are used as indices to the object databases.

In this system ten primitives (block, truncated pyramid, pyramid, bent block, cylinder, truncated cone, cone, truncated ellipsoid, ellipsoid and bent cylinder) are modeled which should be able to model a large number of objects. It is not, however, sufficient to model many real objects, such as trees, animals and clouds.

The aspect hierarchy consists of three levels, based on the faces appearing in the aspect set.

- Aspects consist of the possible groups of faces for each of the primitives. The visible primitives can be identified. However, because of occlusion, some of the faces in an aspect may be partially or completely missing. When this occurs, we may need to analyze the arrangement of the remaining faces, and so second level of the aspect hierarchy is introduced.

- Faces that make up the various aspects form the second level of the aspect hierarchy. Reasoning about the type and arrangement of visible faces can permit identification of an aspect even when it is partially occluded. Due to occlusion, however some of

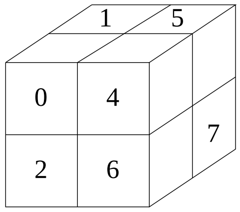

Figure 2.4: Numbering for the eight octants of an object array.

the contours that make up a face may be partially or completely missing. There is the need to find a new solution when this happens.

- Boundary Groups are subsets bounding contours of the faces and make up the third and lowest level of the aspect hierarchy. The boundary group is able to provide a mechanism for identifying the face type even when the face is partially occluded.

For more details on the technique, the interested reader is directed to [8].

One major limitation of the models presented here is that the systems are not able to correctly recover and recognize objects with irregular shapes or objects whose parts of features are occluded (for example, handle of a cup).

2.5 Volumetric models

Meagher [27] and Jackins and Tanimoto [15] have both proposed using octrees to represent three-dimensional objects. Octrees provide a representation method that falls into the *recursive* category. The approach may be considered a three-dimensional extension of quadtree methods. An octree is defined as follows [15]:

We will assume a cube, aligned along a given coordinate system, completely encloses an object to be represented. An octree representation of this object may be derived by successively subdividing in the three dimensions defined by the axes of the coordinate system to form octants. Octants are often referred to by the numbers 0 to 7, though the numbering scheme has not been standardized. Figure 2.4 shows an example enumeration of the suboctants for an octant.

In the representation of an object, an octant may either be full, partially full, or empty. A partially full octant is recursively subdivided into suboctants. Subdivision continues until all octants are either full or empty, or until a predefined cutoff depth is reached. Any partially full node at the cutoff depth is classified as full.

The successive subdivision may be represented as a tree with partially full octants at the internal nodes, and full and empty octants at the leaves. A common convention is to associate the labels with octants as follows:

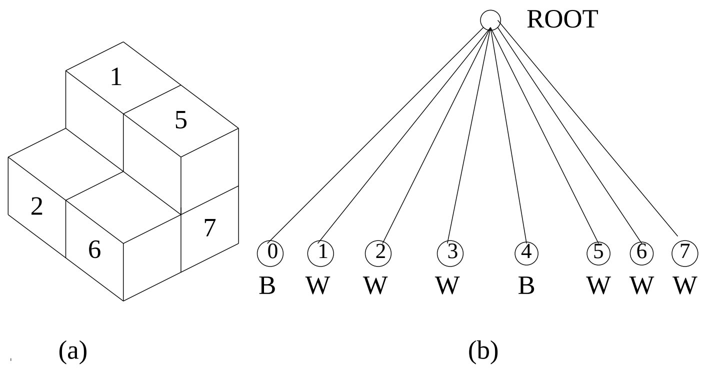

Figure 2.5: (a)A staircase (b)Octree representation of the staircase

- Nodes representing full octants are labeled WHITE or W.
- Nodes representing partially full octants are labeled GREY or G.
- Nodes representing empty octants are labeled BLACK or B.

Figure 2.5 shows the breakdown of a staircase shaped object into octants, and the corresponding octree. The enumeration scheme given in Figure 2.4 was used.

The primary object manipulations for an octree are the Boolean operations (union, intersection, and difference) and the geometric operations of translation, scaling and rotation.

There are several advantages to this data structure [27]. First, there is a single primitive shape, the cube. An arbitrary object can be represented to the precision of the smallest cube. Also, only a single set of manipulation and analysis algorithms is required for all objects. New techniques are not needed to handle more complex or sophisticated shapes.

Operations such as hidden surface display and interference detection show only linear growth because all objects are kept spatially pre-sorted at all times. By traversing the tree in the proper sequence, regions of space will be visited in a uniform direction in space. Thus the hidden-surface display algorithm requires no searching or sorting. The trees representing the objects to be displayed are simply traversed in a specific order, depending on the view direction.

The main disadvantage of this technique is the large memory requirement. Meagher [26] has proven that the memory and processing computation required for a 3D object is of the order of the surface area of the object. Depending on the object and the resolution, this can still represent a very large storage requirement. Several million bytes of node storage may be necessary to represent realistic situations.

2.6 Flow-based surface models

In [36], Szeliski presented the construction of a 3D surface model from image sequences of an object rotating in front of a stationary camera. This approach uses a large number of images with a small amount of motion between successive images. This makes it easier to compute flow, but makes individual flow measurement much less reliable. To compensate for this, an incremental estimation algorithm is used to integrate measurements from successive frames and to reduce the uncertainty over time. The method for creating 3D surface models used the object silhouette to "carve out" a bounding volume for the model [34]. To obtain a more detailed description, optic flow of the texture marks is needed in order to give us a dense estimate of surface shape.

The shape from rotation algorithm developed by Szeliski operates as follows:

1. The 2D optical flow between successive image pairs is extracted over the whole image. The correlation surface corresponding to the Sum of Squared Differences (SSD) measure is used to compute both the best flow estimate at each point and its 2D uncertainty.

2. Using the known object motion, project this flow into a 3D position measurement with an associated 3 x 3 uncertainty at each point. This "cloud" of intensity-tagged depth values is then refined by merging nearby points from successive frames whose uncertainties overlap sufficiently.

3. A locally parameterized surface is then fitted to this collection of points to reduce noise in nearby measurements and to fill in the data where it is unreliable.

For more details on the technique, the interested reader is directed to [35].

This technique gives a relatively good result of the reconstructed 3D surface model. However, this technique is not able to reconstruct the interior parts of the object or objects with irregular shape.

Chapter 3

Previous Work: Stereopsis

If the same object is seen from two different positions (as shown in Figure 3.1) and the camera geometry is known, its three-dimensional position can be measured. This technique is called *stereo vision* or *binocular vision* [32]. It is the most important way in which humans capture short distance range information. In computer vision, stereo vision is also important because it works under a variety of conditions. For example it has been used for obtaining terrain shapes from aerial photographs, where active ranging methods cannot be applied. Another advantage of stereo vision is that it does not require any special lighting.

The main computational problem in stereo vision is finding corresponding points in the left and right images. Once the correspondences are determined, the three-dimensional positions of these points are easily computed from the pairs of points in both images and knowledge of the camera geometry.

Figure 3.1: A pair of stereo images

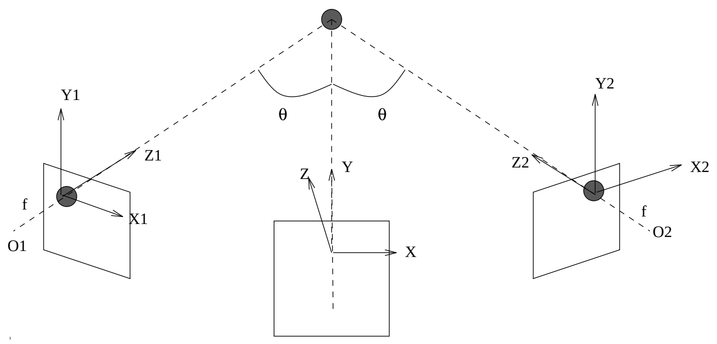

Figure 3.2: Geometry of stereo vision (adapted from [32]

3.1 Stereo Image Geometry

Let us assume that two cameras with the same focal length f are placed so that the distance between the centers of their lenses O_1 and O_2 is d, as shown in Figure 3.2, and the angle between the two optic axes is 2θ. Coordinate systems $X_1 - Y_1 - Z_1$ and $X_2 - Y_2 - Z_2$ are defined for these cameras respectively as shown in the figure. Let us define a coordinate system $X - Y - Z$ so that the Z axis bisects the angle between the Z_1 axis and Z_2 axis as shown in the figure.

Following Shirai [32], the relationship between these coordinate systems is:

$$\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} = \begin{pmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{pmatrix} \begin{pmatrix} X + d/2 - fsin\theta \\ Y \\ Z \end{pmatrix} \tag{3.1}$$

$$\begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \end{pmatrix} = \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \begin{pmatrix} X - d/2 + fsin\theta \\ Y \\ Z \end{pmatrix} \tag{3.2}$$

Suppose that a point p at (X, Y, Z) is observed by two cameras at (x_1, y_1) and at (x_2, y_2) on image planes, 1 and 2 respectively. Assume pinhole camera models with identical focal lengths f. Their relationship is given as follows:

$$x_i = fX_i/(f - Z_i) \tag{3.3}$$

$$y_i = fY_i/(f - Z_i) \tag{3.4}$$

for $i = 1, 2$.

If two corresponding points at (x_1, y_1) and at (x_2, y_2) are obtained in the two image planes (as shown in Figure 3.3), the three-dimensional position (X, Y, Z) of the point can

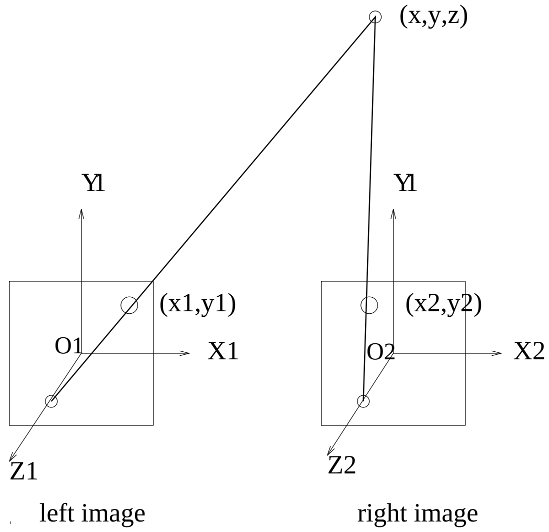

Figure 3.3: Stereo imaging system

be derived from Eqs. (3.1) - (3.4). A point at (x_1, y_1) corresponds to a line of sight that passes through O_1 and (x_1, y_1). The line of sight is observed on the other image plane as a line which is called the *epipolar line*. If θ is 0, for example, the epipolar line is a horizontal line at the same height as the point at (x_1, y_1). If θ is positive, the epipolar line is tilted.

When $\theta = 0$ (in this case, $Z_1 = Z_2 = Z$), then the depth $(f - Z)$ is given as:

$$f - Z = fd/(x_1 - x_2) \tag{3.5}$$

Once Z is determined, X and Y can be easily obtained from Eq. (3.3) and (3.4). Since f and d in Eq. (3.5) are fixed, the depth is inversely proportional to (x_1, x_2), the difference of the x coordinates of the two image planes.

Before getting into details about stereopsis, there are few terms that need to be defined here.

- **Disparity** refers to the angular difference in position of the image element in the two eyes.

- **Distance** refers to the objective physical distance from the viewer to the object, usually measured from one of the two eyes.

- **Depth** refers to the subjective distance to the object as perceived by the viewer, usually measured relative to fixation point or some other three-dimensional point.

If the geometric configuration of two cameras is known, then the disparity encodes the 3D depth of the point [9]. How should the disparity of the point be measured? There are

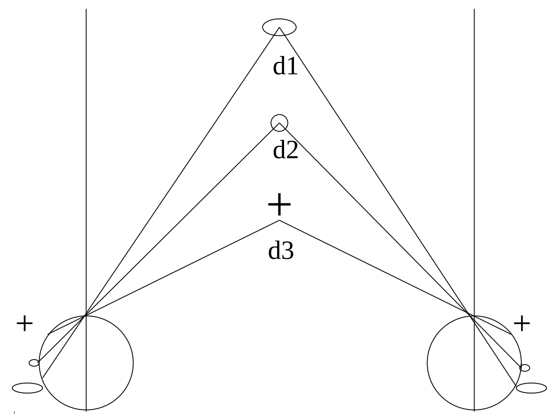

Figure 3.4: **Definition of Disparity** The disparity of a point refers to the angular difference in position of the image element in two eyes. Given a fixation point, such as the circle in the figure, disparities may be crossed (d_3), uncrossed (d_1), or zero (d_2), relative to the fixation point.

many different approaches presented in the literature. The following section will briefly discuss some of these techniques.

3.2 Classic Disparity Detectors

3.2.1 Correspondence

The Correspondence Problem

If two objects are separated in depth from a viewer, then the relative positions of their images will differ in the two eyes. This can be demonstrated in the following manner. Hold your thumb at arms length, and view it first with the left eye closed, then with the right eye closed. You will notice that the background shifts relative to the position of the thumb in the two images. The process of stereo vision, in essence, measures this difference in the relative positions and uses it to compute depth information for visible surfaces in the scene.

As can be seen from Figure 3.4, disparity varies with the relative positions of objects. Suppose the eyes are fixating at a particular point (d_2 in Figure 3.4). Objects in front of the fixation point (d_3 in Figure 3.4), have convergent (or crossed) disparities, objects beyond the fixation point have divergent (or uncrossed) disparities relative to the fixation point (d_1 in Figure 3.4), and objects at the fixation point have zero disparity. As an object moves away from the fixation point, its disparity increases in magnitude. Given the viewing parameters, it is straightforward to derive the relationship between distance and disparity (details for the rotational case are given later in this thesis). The critical problem, however, is how should the disparity be measured? Marr [21] proposed that

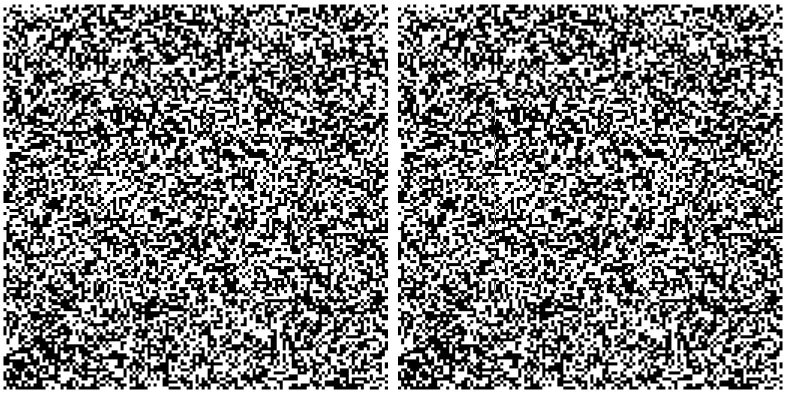

Figure 3.5: Left and right views of random dot stereogram

there are three steps involved:

1. A particular location on a surface in the scene is selected from one image.
2. The same location is found in the other image.
3. The disparity between the two corresponding image points is measured.

When two corresponding image points are identified, the actual computation of distance involves a simple geometric transformation.

The heart of the problem lies in identifying elements in each image which correspond to the same element in the scene.

Determining What To Match

One of the most commonly used stereo examples is that of the three level random dot stereogram as shown in Figure 3.5. This stereogram shown in Figure 3.5 has a surround at zero disparity, a first level at four pixel disparity and a central region at eight pixel disparity. When the stereogram is viewed stereoptically (left to left eye view, right to right eye view), a 3D structure of three layer form of surface is observed. Thus 3D structure can be encoded without the existence of visible 2D features. Random dot stereograms are powerful tools for researchers in the stereo field, as the depth information encoded in the stereo pair is available only from disparity measurements.

Marr and Poggio [24] describe two physical constraints that are relevant to the stereo process:

1. A given point on a physical surface has a unique position in space at any one time.

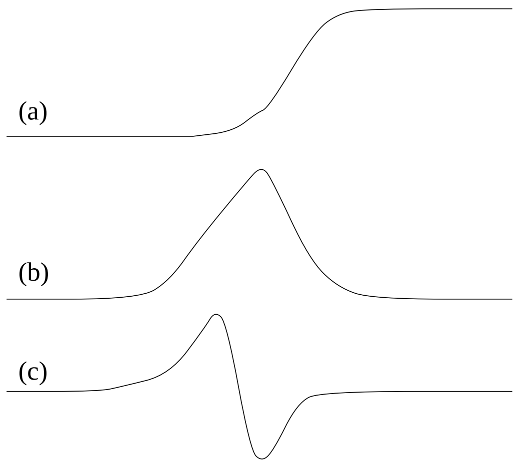

Figure 3.6: **One-Dimensional Problem of Detecting Image Changes** Fig.(a) illustrates a one dimensional step change in irradiance. The change is associated with an extremum in the first derivative (b) and a zero-crossing in the second derivative (c).

2. Matter is cohesive and separated into objects. Furthermore, the surfaces of objects are generally such that the changes in the surface are very small compared with their distance from the viewer.

The stereo matching process must operate from descriptions of images. Thus, we want to match "features" of surfaces in the scene, by way of properties of the images. Suppose we extract an irradiance value from a position in one image. In the second image, we have little hope of locating the position of the corresponding point, since irradiance everywhere is nearly uniform. As a consequence of this difficulty in identifying image locations that accurately reflect specific surface locations, the physical constraints proposed by Marr and Poggio are not directly applicable.

While portions of a uniform surface such as a matte white wall cannot be distinguished, this will not be the case for scratches, texture, or other markings, or for boundaries between surfaces. Such features will give rise to sharp changes in the reflective properties of the surfaces, and will in general cause changes in the irradiances of the image. As a consequence, there should be a direct correspondence between the changes in irradiances and specific surface locations. If we can reliably locate such irradiance changes, then we should be able to apply the two physical constraints to them. Marr and Poggio proposed [24] that the computation of disparity takes place by comparing symbolic descriptions of features in an image that correspond to changes in irradiance, and hence to locations on a surface where some physical property is changing.

How should this variation in irradiance be detected? Consider the case of a one-dimensional irradiance profile. A step change in irradiance is shown in Figure 3.6(a). The first derivative of this profile is shown in Figure 3.6(b), and its second derivative in Figure 3.6(c). It can be seen that the sharp change in irradiance gives rise to an extremum in the first derivative and a zero-crossing in the second derivative. When we turn to two-dimensional arrays of irradiance, such as an image, the same observations hold true. Thus, Marr and Poggio [24] were the first to suggest the use of zero-crossings, detected in the second derivative of the images as monocular features in stereopsis. Later, many other researchers (for example, Mayhew and Frisby and Gale [25] and Grimson [12]) have expanded this theory through modifying the types of filtering and features extracted. However, the basic computational concept has remained unchanged [19]: a spatial-frequency band-pass filter is applied to the image, and a number of features are extracted from the result. Current existing algorithms select either zero-crossings, or zero-crossings and peaks, at a number of different spatial frequencies as their monocular primitives.

In some existing stereo algorithms, such as Grimson's [12] implementation of Marr and Poggio's algorithm, a rotationally symmetric operator was chosen. The image was initially convolved with a Gaussian function:

$$G_\sigma * E(x,y) \tag{3.6}$$

where the Gaussian is given by

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\pi\sigma^2}} \tag{3.7}$$

and E is the image intensity function. The Gaussian is a band-limited operator. This means that it is sensitive to a particular range of spatial frequencies. When a Gaussian operator is applied to the image, the image will be blurred by an amount dependent upon the value of σ [12]. The application of a large Gaussian will tend to enhance gross features of the image, while the application of a small Gaussian will tend to enhance fine features of the image. One can obtain descriptions of the image with varying degrees of resolution by filtering the image with a series of $\bigtriangledown^2 G$ operators, each with a different sized Gaussian.

In Grimson's implementation, the Laplacian $\bigtriangledown^2$ is then applied to the filtered image and the zero crossings are then located. The Laplacian is given by:

$$\bigtriangledown^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{3.8}$$

From the zero crossings obtained with this symmetric convolution, orientations are determined.

Although there are many possible ways of filtering the image at different degrees of coarseness, Marr and Hildreth [23] suggest that the Gaussian is one of the best filters to use. In particular, they propose that the image should be filtered by convolution with a series of different sized $\bigtriangledown^2 G$ operators, and the zero-crossings of each output be located and matched Interocularly. Two features of zero-crossings can be computed. One is the sign, or contrast, of the zero-crossing. This can be computed simply by noting if the convolution values change from positive to negative or negative to positive while scanning from left to right across the zero-crossing. The second feature is the local orientation of a zero-crossing on the image plane. This can be found by computing the gradient of the convolution values across the zero-crossing and taking the orientation of the projection of the gradient onto the image as the orientation of the zero-crossing.

Mayhew and Frisby [25] have shown that even though the Laplacian of the Gaussian seems useful for the first step, the process of obtaining zero crossings is questionable. They have shown that in certain cases an incorrect depth will be recovered if zero crossings are used alone. They argue that (at least) the peaks in the Laplacian of the Gaussian should be also used along with zero crossings.

Beside the Laplacian of Gaussian filter, many other filters also have band-pass properties, but with equal or better localization properties. For instance, Canny [5], or Gabor [10] edge filters could be used.

Marr and Poggio [24] note that the difficulty of the correspondence problem and the subproblem of eliminating false targets is directly proportional to the range and resolution of disparities considered and to the density of matchable features in an image. In other words, the greater the range of disparity over which a match is sought, the greater the number of false targets. At the same time, the finer the resolution at which features are detected, the greater the density of features and thus the greater the number of potential false targets. Thus, one can avoid the false targets problem either by reducing the range of the disparities accepted or by reducing the resolution of the disparities accepted.

There are several factors related in the choice of suitable image primitives. It is important to use features that can be expected to produce reliable information about surface structure once they have been matched correctly. It is also important that matches occur sufficiently often to provide a fairly dense description of the disparity. Note that the density of any measurement will be limited by the surface texture. Smooth surfaces will not give rise to as rich a description as surfaces with complex textures.

The number of possible matches in a given region increases polynomially with the density of a given symbolic feature [19]. Therefore, finding the correct match will become more difficult as the density grows.

3.2.2 Correlation

Under a correlation based approach, it is assumed that the left and right images are simply shifted versions of each other, and that the measurement of this shift is equivalent to determining the stereo parallax. Here, it is assumed that the correct shift is the shift that sets the integral of the square of the difference between the two images to zero. This sum has a minimum when the images are shifted the correct amount with respect to each other. In a corresponding way, the correlation (the integral of their product) has a maximum. One can generate this idea to measure the shift between two regions in the image. The disparity or shift that will be assigned to the point x can be written as [19]

$$max_{disp} \int W(z) left(x + disp + z) right(x + z) dz, \tag{3.9}$$

where W is a windowing function used to localize the disparity measure, and left and right are the left and right images.

Unfortunately, the left and right images are not simply shifted views of each other as Horn noted in [13]. For large base-height ratios in hilly terrain, or in built-up areas, some surfaces may be seen only in one image. Those that are seen in both will appear with different foreshortening in the two images; that is, the surfaces appear compressed along one dimension to differing degrees. In practice, correlation is not able to deal with scenes that are distorted by perspective projection or by interocular differences in illumination [13]. The same surface patch will appear with different brightness when viewed from two different directions. Specular reflection in particular will pose a difficult problem. The task of determining a unique minimum can be very difficult when there are large regions of near constant intensity. On the other hand, there are problems when there is a great deal of variation in intensity.

As a technique for measuring disparity, correlation has difficulties in dealing with [17]

1. perspective projection.
2. interocular illumination differences.
3. disparity gradients.
4. too much structure in an image.
5. too little structure in an image.

By adding preconvolution steps, or blurring windowing functions, a number of problems may be reduced. However, the underlying theory becomes lost in the use of image and correlation space operators whose functions are at best of a "heuristic" nature.

3.2.3 Summary

Correlation and correspondence approaches to stereopsis are in a sense complementary. Correlation produces a dense set of responses but has difficulty with constant or rapidly changing structure and with interocular image differences, while correspondence avoids some of these problem by considering the image at different scales but then fails to obtain a dense set of responses by matching only sparse tokens. Therefore, a different technique is needed to avoid the problem of structure at different scales and at the same time produce a dense response. A brief description of such a technique is in the next section.

3.3 The Local Phase Based Stereo Algorithm

In the past, the task of recovering surface height from two images of a scene has been usually expressed as either a correlation or correspondence process [19]. Even though both methods have given some limited success, neither method has reached the performance associated with the human visual system.

More recently, a number of researchers have proposed that stereo (and motion and form) processing should be considered as the task of measuring the local phase difference between bandpass filtered version of the monocular signals. This new disparity measurement technique expresses the disparity in terms of phase differences in the output of local, bandpass filters applied to the left and right views [16,31,20].

The main advantage of this local phase-based approaches is that disparity estimates are obtained with subpixel accuracy, without requiring explicit subpixel signal reconstruction or subpixel feature detection and localization. Since there are no restrictions to specific values of phase (e.g., zeros) that must first be detected and localized, the density of measurements is also expected to be high [9].

3.3.1 Phase-difference techniques

The local phase-based disparity measurement technique that used in this thesis is based on the algorithm proposed by Fleet, Jepson and Jenkin [9]. It is described as follows: A prefiltering stage based on Gabor functions [10] is used to localize signal energy in both space and frequency. Let $R_l(x, k_0)$ and $R_r(x, k_0)$ be the left and right views after convolution with a Gabor kernel, which is given by

$$Gabor(x; \sigma, k_0) \equiv e^{ixk_0} G(x; \sigma), \tag{3.10}$$

where $G(x; \sigma)$ is a Gaussian window with standard deviation σ, and k_0 is the peak tuning frequency (in radians) to which the filter responds maximally. The amplitude spectrum of (3.10) is Gaussian-shaped and centered at k_0. Then $R_l(x, k_0)$ and $R_r(x, k_0)$ can be written as

$$\begin{aligned} R_l(x, k_0) &= \rho_l(x) e^{i[k_0 x + \psi_l(x)]}, \\ R_r(x, k_0) &= \rho_r(x) e^{i[k_0 x + \psi_r(x)]}, \end{aligned} \tag{3.11}$$

where $\rho(x)$ and $\phi(x) \equiv [k_0 x + \psi(x)]$ are the amplitude and the phase components of response. Here, (3.11) can be also written as

$$R(x, k_0) = M(x, k_0) e^{ik_0 x}, \tag{3.12}$$

where $M(x, k_0) \equiv \rho(x) e^{i\psi(x)}$ is a lowpass signal. Following Whitham [40] or Papoulis [28], the local frequency can be defined as the spatial derivative of the phase signal, $k(x) \equiv \phi'(x)$. If the phase of $M(x, k_0)$ is linear, i.e., $\psi(x) = k_1 x$, then $R(x; k_0)$ is just an amplitude-modulated sinusoid with constant frequency $k_0 + k_1$. Otherwise, the phase derivative $\phi'(x) = k_0 + \psi'(x)$ provides a local, constant-frequency approximation to $R(x; k_0)$.

The local image disparity, at a specific position x, for an initial guess d_i, is then defined to be the shift $d(x)$ such that

$$\phi_l(x - d(x)/2) = \phi_r(x + d(x)/2), \tag{3.13}$$

and $\mid d(x) - d_i \mid$ is as small as possible [9].

If the initial guess is sufficiently good and the filter outputs are sinusoidal with constant frequency k_0, then the shift to match the phases of the left and right filter outputs is given by

$$\tilde{d}_0(x) \equiv [\phi_l(x) - \phi_r(x)]_{2\pi} / k_0, \tag{3.14}$$

where $[\theta]_{2\pi}$ denotes the principal part of θ, that is, $[\theta]_{2\pi} \in (-\pi, \pi]$. However, if the outputs are not sinusoidal with constant frequency k_0, then the disparity estimates provided by $\tilde{d}_0$ will not be exact. As Fleet notes, this solution can be improved by replacing k_0 in (3.14) by the average local frequency between the left and right views. This gives:

$$\begin{array}{l} \tilde{d}_l(x) = [\phi_l(x) - \phi_r(x)]_{2\pi} / \overline{k}(x), \\ \text{where } \overline{k}(x) = \frac{1}{2}(\phi_l'(x) + \phi_r'(x)). \end{array} \tag{3.15}$$

Then, $\tilde{d}_l(x)$ gives the exact disparity when the left and right images both have locally constant frequencies, but not necessarily equal to the filter tuning k_0. When the left and right outputs are shifted versions of one another, but do not have constant frequency, the disparity error is:

$$\varepsilon(x) = O(\delta^2 \phi'''(x) / \tilde{\phi}(x)) \tag{3.16}$$

When the disparity estimates are not exact, then shifting the signals by $\tilde{d}_l(x)$ will not precisely match the phases in the left and right filter outputs as required by (3.13). Since the error depends on the initial disparity, iterating the basic measurements can be used to improve the accuracy of disparity estimates. During the current iteration the images are shifted according to the current disparity approximation, the local image disparity between the shifted left and right signals is computed using either $\tilde{d}_0$ or $\tilde{d}_1$, and the disparity approximation is updated with the result. That is, with an initial guess d_i, the disparity approximation at iteration $t+1$ is given by

$$\tilde{d}^{t+1}(x) = \tilde{d}^t(x) + \Delta\tilde{d}^t(x), \tag{3.17}$$

where

$$\tilde{d}^0(x) = d_i, \quad \Delta\tilde{d}^t(x) = ([\phi_l(x - (\tilde{d}^t(x)/2))]_{2\pi} - [\phi_r(x + (\tilde{d}^t(x)/2))]_{2\pi})/k(x),$$

and $k(x)$ is given by $\overline{k}(x)$ in (3.15) for the new predictor, or by k_0 for the old predictor (3.14). The estimate $\tilde{d}^t(x)$ converges to the exact disparity, as defined by (3.13).

3.3.2 Phase stability and scale-space singularities

The transformation between the left and right stereo images is often assumed to be approximated by image translation (i.e., $I_l(x) = I_r(x+d)$). Fleet, Jepson and Jenkin [9] argue that an affine transform,

$$I_l(a(x)) = I_r(x), \tag{3.18}$$

where $a(x) = a_0 + a_1 x$, is a more realistic first approximation. If this is assumed, then because the filters have constant bandwidths, the two output signals R_l and R_r will satisfy

$$R_l(a(x), k_1) = R_r(x, k_2), \tag{3.19}$$

where $k_2 = k_1 a_1$. This means, a rescaling of the input can be matched by changing the peak tuning frequency so that the same image structure is extracted by the filters. It was attempted to recover $a(x)$ from the left and right responses of the same filter, from $R_l(x, k_1)$ and $R_r(x, k_1)$. For this to succeed, the image property used for binocular matching must be *stable* under small perturbations [9].

Following Fleet [9], let the scale-space expansion of $I(x)$ be given by

$$S(x, \lambda) = Gabor(x; \sigma(\lambda), k(\lambda)) * I(x), \tag{3.20}$$

where $*$ denotes convolution, and λ is the scale parameter. The peak tuning frequency of the filter is given by

$$k(\lambda) = 2\pi/\lambda, \tag{3.21}$$

Following Fleet, the approximate radius of spatial support is given by

$$\sigma(\lambda) = \frac{1}{k(\lambda)}\left(\frac{2^\beta + 1}{2^\beta - 1}\right). \tag{3.22}$$

where β is the bandwidth of the Gabor in octaves, and the amplitude and phase components of $S(x,\lambda)$ are given by

$$\begin{aligned} \rho(x,\lambda) &= |S(x,\lambda)| \equiv \sqrt{(Re[S(x,\lambda)])^2 + (Im[S(x,\lambda)])^2}, \\ \phi(x,\lambda) &= arg[S(x,\lambda)] \equiv Im[log_e S(x,\lambda)]. \end{aligned} \tag{3.23}$$

As Fleet, Jepson and Jenkin presented in their paper [9], the phase structure is generally stable with respect to scale perturbations. They also indicated that in some regions phase *does* depend significantly on scale and a major source of instability is the occurrence of singularities in the phase signal.

Let (x_0, λ_0) denote the location of a singularity. The neighborhoods just above singular points (i.e. for $\lambda > \lambda_0$) are characterized by local frequencies that are significantly below the corresponding peak frequencies $k(\lambda)$. Below singular points (i.e., for $\lambda < \lambda_0$) the neighborhoods are characterized by local frequencies that are significantly higher than corresponding peak tuning frequencies. In addition, the local frequency of the response changes rapidly as a function of spatial location, with local maxima that roughly coincide with the singular locations. As Fleet, Jepson and Jenkin note this rapid variation in phase behavior is not a source of instability with respect to scale, it will however cause problems for disparity computation that is based on a constant-frequency model of the local filter output as in (3.15). The numerical approximation of $\phi'(x)$ for use in (3.15) will be poor, the local domains of convergence will be extremely small, and the convergence will be slow.

Therefore, it is necessary that singularity neighborhoods are detected so that incorrect disparity estimates can be avoided. Here, Fleet, Jepson and Jenkin [9] proposed the following:

To detect the neighborhoods above and below the singular points, a constraint on local frequency in relation to the peak tuning frequency was considered. That is, the distance between the local frequency of response $\phi_x(x,\lambda)$ and the peak frequency $k(\lambda)$ should be constrained. This can be expressed as a function of the standard deviation of the amplitude spectrum $\sigma_k(\lambda)$.

$$\frac{|\phi_x(x,\lambda) - k(\lambda)|}{\sigma_k(\lambda)} < \tau_k, \qquad \sigma_k(\lambda) = k(\lambda)\left(\frac{2^\beta - 1}{2^\beta + 1}\right). \tag{3.24}$$

The neighborhoods to the left and right of singular points can be detected with a constraint on local amplitude variation,

$$\sigma(\lambda)\frac{|\rho_x(x,\lambda)|}{\rho(x,\lambda)} < \tau_\rho, \tag{3.25}$$

where $\sigma(\lambda)$ defines the radius of the filter support (3.22). If $\rho(x_1,\lambda_0)/\,|\rho_x(x_1,\lambda_0)|$ is taken to be an approximation to the distance from a singularity, then (3.25) detects points that lie within a distance of $\sigma(\lambda)/\tau_\rho$ from a singularity. Furthermore, as τ_k and τ_ρ decrease, the constraints become tighter and larger neighborhoods are detected.

Fleet, Jepson and Jenkin [9] have developed an algorithm which incorporate the above disparity predictor and the removal of singularity neighborhoods. In which, a coarse-to-fine control strategy is used. The Gabor filters had bandwidths of one octave, and the different wavelengths to which the filters were tuned were 4, 8, 16 and 32 pixels. The coarsest scale begins computation with initial guesses of $d_i = 0$. All computed disparities at one level are then passed down as initial guesses for the next finer level. Moreover, these raw measurements are then used as the data points for a membrane based regularization step which builds local surface support. Breaks are fitted into the local surface whenever the disparity gradient (measured in either horizontal or vertical direction) exceeds a particular threshold which is set based on the underlying frequency and pass region of the channel used.

These new techniques have shown some improvements over earlier algorithms. They provide a rich and dense set of disparity measures over a large class of image textures. The measurements provided are ideally suited to our requirements. Thus, in the work presented here we utilize a stereopsis algorithm based on a new improved version of the above local phase difference technique developed by Jenkin and Jepson [18] to measure the disparity between a pair of images. In this thesis, this new algorithm is used. Filters were tuned with 3, 6, 12, 24, and 48 pixels in 3 different orientations, 0^0, 45^0, and -45^0. These filters are interocular-oriented matched to the same orientation surface with breaks, where $\tau_k = 1.2,\ \tau_\rho = 1$.

Chapter 4

Framework Overview

4.1 Introduction

The shape from rotation algorithm developed in this paper converts a series of images into a 3D model of the object. The algorithm operates by integrating repeated views of the object as it undergoes a controlled rotation about a known axis. Each pair of images is treated as a stereo pair and the disparity computed from the stereo algorithm is used as the primitive measurement of the object. In order to map disparities into an object centered coordinate system, the experimental setup must be calibrated. One approach to calibration is to image a known 3D reference model, such as a cube, and to use the known position of points on the object to calibrate the system. In order to automatically determine the object's rotation angle without the use of any additional sensors, a rotational stage which gives a precise measure of the rotation was used. A black background was placed behind the turntable so as to reduce the effect of any background texture.

The actual shape from rotation using stereo algorithm operates in the following stages.

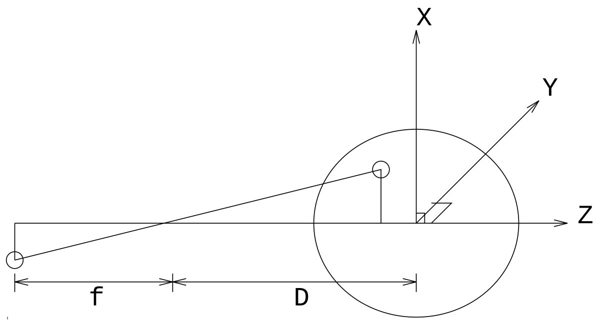

Figure 4.1: Calibrating Camera System

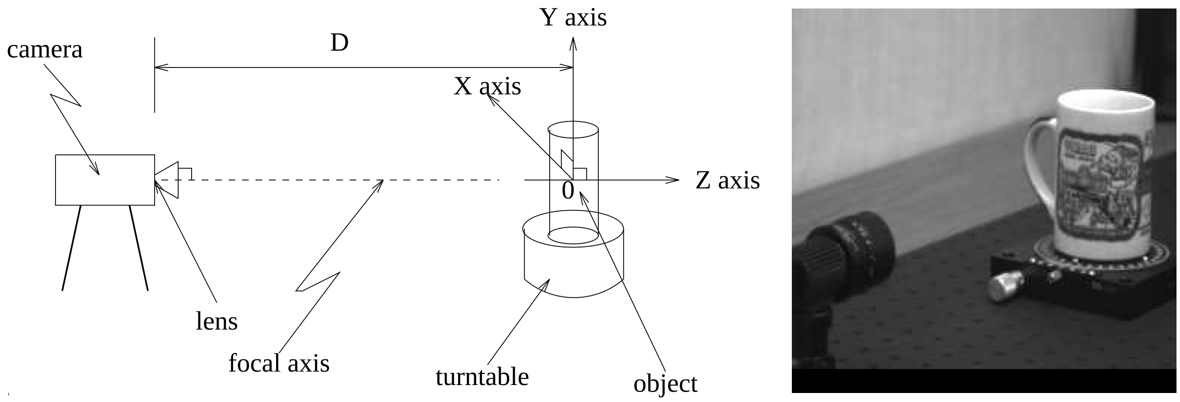

Figure 4.2: Experimental set up

1. The camera system is calibrated. The transformation between the camera image coordinates and the 3D world coordinate system is computed through a known 3D model such as a cube.

2. The disparity between successive image pairs is extracted using an existing stereo algorithm. The first image is considered as the left image and the second image as the right image for the first processing pair. In the next iteration the second image is the left image and the third image is the right image for the second processing pair, and so on.

3. After running the stereo algorithm on a pair of images, the disparity of the image points are obtained. As the camera system has been calibrated, it is possible to compute the true depth of each recovered point in the image (this is described in a later section).

4. The recovered points (x,y,z) of the image are represented in an octree.

5. Go back to step 2, repeat the processing for a complete 360^o rotation of the object.

4.2 Camera Calibration (after Horn[14])

The experimental set up of the shape from rotation experiment is shown as in Figure 4.2. The camera is positioned in front of the turntable with the camera's focal axis roughly aligned with the Z axis of the turntable. An object, such as a cup, is put on the center of the turntable. An object based coordinate system is centered at the top of the rotation platform, with the Y axis pointing up from the platform, and X and Z lying on the rotation plane. A calibration grid on the turntable defines the X and Z directions. The origin of the object based coordinate system is at the intersection of camera focal axis and XY plane.

To recover the transformation matrix between the camera image coordinates and the rotational stage center in world coordinate system, we use a known model such as a cube to calibrate the system. There are many different techniques available for calibrating a camera system [37,14]. One method is to discover the actual camera model parameters that relate the two coordinate systems [11]. Following Horn[14], a calibration object is used to experimentally determine the transformation from the object to image coordinates.

Assume a pin-hole camera model as shown in Figure 4.2. Let (u, v) be the image of a point (x_c, y_c, z_c) in the camera coordinate system and let f be the focal length of the camera. Furthermore, let (x_a, y_a, z_a) be the coordinate of the point in the rotating stage coordinate system.

Points (x_c, y_c, z_c) and (x_a, y_a, z_a) are related as follows:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_a \\ y_a \\ z_a \\ 1 \end{bmatrix} \tag{4.1}$$

Where the r_{ij} elements define the transform in homogeneous co-ordinates to map one coordinate system to the other.

In an ideal case (u, v) is related to (x_c, y_c, z_c) by $(u/f = x_c/z_c,\ v/f = y_c/z_c)$. In an electro-optical system, it may be very difficult to ensure the two axes of the measurement system are exactly orthogonal and the distances measured along the axes are scaled equally. In order to take these effects and others into account, a more general model of the camera imaging system is required. One approach [14] is to assume that an affine transform of the image occurs in the camera. An *affine* transformation in the plane is linear and can work for uniform scaling, translation, rotation, skewing, and shearing. This transformation can absorb all of the departures from the ideal situation as follows:

- Scaling error, due to inaccurate knowledge of focal length (f).
- Translation error, due to inaccurate knowledge of the origin.
- Rotation error, due to inaccurate knowledge of image sensor rotation.
- Skewing error, due to departures from orthogonality in the sensor.
- Shearing error, due to unequal scaling in the image axes.

Note that this does not deal with all possible image distortions, as radial distortions of the camera are not modeled, for example. Assuming that the calibration can be modeled as an affine transformation, then the imaging process can then be modeled as:

$$\begin{bmatrix} u/f \\ v/f \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c/z_c \\ y_c/z_c \\ 1 \end{bmatrix} \tag{4.2}$$

so that

$$\begin{aligned} u/f &= (a_{11}x_c + a_{12}y_c + a_{13}z_c)/(z_c) \\ v/f &= (a_{21}x_c + a_{22}y_c + a_{23}z_c)/(z_c) \end{aligned} \tag{4.3}$$

Combining Eq. 4.1 with Eq. 4.3, results in:

$$\begin{aligned} \frac{u}{f} &= \frac{a_{11}(r_{11}x_a+r_{12}y_a+r_{13}z_a+r_{14})+a_{12}(r_{21}x_a+r_{22}y_a+r_{23}z_a+r_{24})+a_{13}(r_{31}x_a+r_{32}y_a+r_{33}z_a+r_{34})}{(r_{31}x_a+r_{32}y_a+r_{33}z_a+r_{34})} \\ \frac{v}{f} &= \frac{a_{21}(r_{11}x_a+r_{12}y_a+r_{13}z_a+r_{14})+a_{22}(r_{21}x_a+r_{22}y_a+r_{23}z_a+r_{24})+a_{23}(r_{31}x_a+r_{32}y_a+r_{33}z_a+r_{34})}{(r_{31}x_a+r_{32}y_a+r_{33}z_a+r_{34})} \end{aligned} \tag{4.4}$$

Simplifying this above equation, gives:

$$\begin{aligned} \frac{u}{f} &= \frac{x_a(a_{11}r_{11}+a_{12}r_{21}+a_{13}r_{31})+y_a(a_{11}r_{12}+a_{12}r_{22}+a_{13}r_{32})+z_a(a_{11}r_{13}+a_{12}r_{23}+a_{13}r_{33})+(a_{11}r_{14}+a_{12}r_{24}+a_{13}r_{34})}{(r_{31}x_a+r_{32}y_a+r_{33}z_a+r_{34})} \\ \frac{v}{f} &= \frac{x_a(a_{21}r_{11}+a_{22}r_{21}+a_{23}r_{31})+y_a(a_{21}r_{12}+a_{22}r_{22}+a_{23}r_{32})+z_a(a_{21}r_{13}+a_{22}r_{23}+a_{23}r_{33})+(a_{21}r_{14}+a_{22}r_{24}+a_{23}r_{34})}{(r_{31}x_a+r_{32}y_a+r_{33}z_a+r_{34})} \end{aligned} \tag{4.5}$$

Defining

$$s = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ s_5 & s_6 & s_7 & s_8 \\ s_9 & s_{10} & s_{11} & s_{12} \end{bmatrix} =$$

$$\begin{bmatrix} a_{11}r_{11}+a_{12}r_{21}+a_{13}r_{31} & a_{11}r_{12}+a_{12}r_{22}+a_{13}r_{32} & a_{11}r_{13}+a_{12}r_{23}+a_{13}r_{33} & a_{11}r_{14}+a_{12}r_{24}+a_{13}r_{34} \\ a_{21}r_{11}+a_{22}r_{21}+a_{23}r_{31} & a_{21}r_{12}+a_{22}r_{22}+a_{23}r_{32} & a_{21}r_{13}+a_{22}r_{23}+a_{23}r_{33} & a_{21}r_{14}+a_{22}r_{24}+a_{23}r_{34} \\ r_{31} & r_{32} & r_{33} & r_{34} \end{bmatrix} \tag{4.6}$$

then Eq. 4.5 can be rewritten as:

$$\begin{aligned} u/f &= (s_1x_a + s_2y_a + s_3z_a + s_4)/(s_9x_a + s_{10}y_a + s_{11}z_a + s_{12}) \\ v/f &= (s_5x_a + s_6y_a + s_7z_a + s_8)/(s_9x_a + s_{10}y_a + s_{11}z_a + s_{12}) \end{aligned} \tag{4.7}$$

Given (x_a, y_a, z_a, u, v) and the camera focal length f, the problem is to determine the twelve unknowns of the transformation s. One way to proceed is to use a number of calibration points whose coordinates are known in the external system. The transformation (s) can then be recovered if we measure the coordinates of the corresponding points in the image, i.e., (x_a, y_a, z_a, u, v). Reformulating (4.7), each measurement results in one pair of linear equations:

$$\begin{aligned} u(s_9x_a + s_{10}y_a + s_{11}z_a + s_{12}) - f(s_1x_a + s_2y_a + s_3z_a + s_4) = 0 \\ v(s_9x_a + s_{10}y_a + s_{11}z_a + s_{12}) - f(s_5x_a + s_6y_a + s_7z_a + s_8) = 0 \end{aligned} \tag{4.8}$$

Six image measurements are required to solve for the twelve unknowns. However, these equations are homogeneous. In other words the transformation is not affected by the uniform scaling of all of the coefficients so that there are actually eleven degrees of freedom. Without any further assumptions, six image measurements will be more than enough to solve the equations, in the absence of noise.

The two constraints induced by a single image measurement can be written in the form $\mathbf{a_i} \cdot \mathbf{c} = \mathbf{0}$ and $\mathbf{b_i} \cdot \mathbf{c} = \mathbf{0}$, where

$$\begin{aligned} &\mathbf{a_i} = (-fx_a, -fy_a, -fz_a, -f, 0, 0, 0, 0, ux_a, uy_a, uz_a, u), \\ &\mathbf{b_i} = (0, 0, 0, 0, -fx_a, -fy_a, -fz_a, -f, vx_a, vy_a, vz_a, v), \\ &\mathbf{c} = (s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12})^T. \end{aligned} \tag{4.9}$$

Given six image measurement vectors $\mathbf{a}_i$, and $\mathbf{b}_i$, it is possible to solve for the vector $\mathbf{c}$. In practice, additional measurements should be used to improve accuracy. It is then no longer possible to find a set of coefficients that will make all of the constraint equations exactly equal to zero. Instead, approximate solutions such as minimizing the sum of squares of the errors are possible,

$$\sum_{i=1}^{n} (\mathbf{w}_i \cdot \mathbf{c})^2, \tag{4.10}$$

where the $\mathbf{w}_i$ are vectors obtained from the measurements. The trivial solution is when all coefficients are zero, that is $\mathbf{c} = 0$. To avoid this trivial solution, additional constraints are needed to be imposed, such as $|\mathbf{c}| = 1$. This is a nonlinear problem.

On the other hand, instead of imposing a nonlinear constraint (as suggested in the previous paragraph) the least square minimization problem can be solved when one of the coefficients, (s_{12}, for example), is fixed in value, (say $s_{12} = 1$). This method is chosen in this thesis since it is easier to solve. The technique is briefly described as follows,

- Read in measurements of all image points $(x_i, y_i, z_i, u_i, v_i)$.
- Calculate all corresponding vectors $\mathbf{a}$ and $\mathbf{b}$ of all image measuring points, and store them in a matrix $\mathbf{w}$ where $\mathbf{w}$ is constructed as follows:

$$\mathbf{w} = \begin{bmatrix} \mathbf{a_1} \\ \mathbf{b_1} \\ \mathbf{a_2} \\ \mathbf{b_2} \\ . \\ . \\ \mathbf{a_n} \\ \mathbf{b_n} \end{bmatrix} \tag{4.11}$$

- To solve in a least squares since $\mathbf{wc} = \mathbf{0}$, we need to solve the following linear system:

$$\begin{bmatrix} -fx_1 & -fy_1 & -fz_1 & -f & 0 & 0 & 0 & 0 & u_1x_1 & u_1y_1 & u_1z_1 & u_1 \\ 0 & 0 & 0 & 0 & -fx_1 & -fy_1 & -fz_1 & -f & v_1x_1 & v_1y_1 & v_1z_1 & v_1 \\ -fx_2 & -fy_2 & -fz_2 & -f & 0 & 0 & 0 & 0 & u_2x_2 & u_2y_2 & u_2z_2 & u_2 \\ . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \\ -fx_n & -fy_n & -fz_n & -f & 0 & 0 & 0 & 0 & u_nx_n & u_ny_n & u_nz_n & u_n \\ 0 & 0 & 0 & 0 & -fx_n & -fy_n & -fz_n & -f & v_nx_n & v_ny_n & v_nz_n & v_n \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ . \\ . \\ . \\ s_{11} \\ 1 \end{bmatrix} = \mathbf{0} \tag{4.12}$$

- Furthermore, the above linear system can be written as $\mathbf{w}'\mathbf{c}' = \mathbf{y}$, where

$$\mathbf{w}' = \begin{bmatrix} -fx_1 & -fy_1 & -fz_1 & -f & 0 & 0 & 0 & 0 & u_1x_1 & u_1y_1 & u_1z_1 \\ 0 & 0 & 0 & 0 & -fx_1 & -fy_1 & -fz_1 & -f & v_1x_1 & v_1y_1 & v_1z_1 \\ -fx_2 & -fy_2 & -fz_2 & -f & 0 & 0 & 0 & 0 & u_2x_2 & u_2y_2 & u_2z_2 \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ -fx_n & -fy_n & -fz_n & -f & 0 & 0 & 0 & 0 & u_nx_n & u_ny_n & u_nz_n \\ 0 & 0 & 0 & 0 & -fx_n & -fy_n & -fz_n & -f & v_nx_n & v_ny_n & v_nz_n \end{bmatrix},$$

  and

$$\mathbf{c}' = [s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}]^T, \qquad \mathbf{y} = [-u_1, -v_1, -u_2, ..., -u_n, -v_n]^T$$

  then $\mathbf{c}' = (\mathbf{w}'^T\mathbf{w}')^{-1}\mathbf{w}'^T\mathbf{y}$ is the least squares solution [1]. That is, the unknown vector $\mathbf{c}'$ can now be recovered provide $(\mathbf{w}'^T\mathbf{w}')$ is non singular. In other words, the transformation s (4.6) can now be solved.

Our next step should be to solve a_{ij} and r_{ij}. From (4.6), we can form 12 linear equations in order to solve a_{ij} and r_{ij}, and there are 18 unknowns to be solved. It is not possible to solve for a_{ij} and r_{ij} due to lack of information about rotation, translation and projection. However, given vector $\mathbf{c}$, we are able to compute the depth of a given image point (u, v) and its true 3D point (x_a, y_a, z_a) without knowing a_{ij} and r_{ij}. This is presented in the following section.

4.3 Recovering Depth

Once the unknown vector **c** has been recovered, the depth of an image point and its true 3D location (x_a, y_a, z_a) can be computed as follows: Assume that an image point (x_a, y_a, z_a) is projected at time t to (u_1, v_1), and that at time $t+1$ it is projected to (u_2, v_2) (after the platform rotates by an amount $\Delta\theta$). Then, $u_2 = u_1 + disp$, where $disp$ is the horizontal component of the disparity obtained using the stereo algorithm. Given s_1 to s_{11}, f, (u_1, v_1), (u_2, v_2), and $\Delta\theta$, we can solve for the unknown point (x_a, y_a, z_a), by solving the following linear equations.

$$\begin{aligned} u_1/f &= (s_1 x_a + s_2 y_a + s_3 z_a + s_4)/(s_9 x_a + s_{10} y_a + s_{11} z_a + 1), \\ v_1/f &= (s_5 x_a + s_6 y_a + s_7 z_a + s_8)/(s_9 x_a + s_{10} y_a + s_{11} z_a + 1), \\ u_2/f &= (s_{13} x_a + s_2 y_a + s_{14} z_a + s_4)/(s_{15} x_a + s_{10} y_a + s_{16} z_a + 1). \end{aligned} \tag{4.13}$$

where

$$\begin{aligned} s_{13} &= s_1 cos\Delta\theta - s_3 sin\Delta\theta, \\ s_{14} &= s_3 cos\Delta\theta + s_1 sin\Delta\theta, \\ s_{15} &= s_9 cos\Delta\theta - s_{11} sin\Delta\theta, \\ s_{16} &= s_{11} cos\Delta\theta + s_9 sin\Delta\theta. \end{aligned} \tag{4.14}$$

The 3D point (x_a, y_a, z_a) can then be found by solving the following linear system:

$$\begin{bmatrix} u_1 s_9 - s_1 f & u_1 s_{10} - s_2 f & u_1 s_{11} - s_3 f \\ v_1 s_9 - s_5 f & v_1 s_{10} - s_6 f & v_1 s_{11} - s_7 f \\ u_2 s_{15} - s_{13} f & u_2 s_{10} - s_2 f & u_2 s_{16} - s_{14} f \end{bmatrix} \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} = \begin{bmatrix} f s_4 - u_1 \\ f s_8 - v_1 \\ f s_4 - u_2 \end{bmatrix} \tag{4.15}$$

The technique to compute this 3D point from the above linear system is as follows:

- An approximation to the focal length f of the pinhole camera is given.
- From the previous section, s_1 to s_{11} are computed.
- Given a rotational angle θ, calculate s_{12}, s_{13}, s_{14}, and s_{15}.
- Since (4.15) is a linear system of the form $\mathbf{Ax} = \mathbf{b}$, where matrix $\mathbf{A}$ and vector $\mathbf{b}$ are given, then we can use LU decomposition to solve for vector $\mathbf{x}$. That is the 3D point (x_a, y_a, z_a) can be computed.

Obviously, z_a is the depth of the point relative to the rotation stage. Based on these 3D points, it is possible to reconstruct the 3D model after a 360^o rotation.

4.4 Representing 3D objects

An octree is used to represent the 3D points that are recovered. A rectangular octree is a regular cellular decomposition of the object space (universe) [6,30]. The universe is subdivided into eight cells of equal size. If any one of the resulting cells is homogeneous, (meaning that it lies entirely inside or outside the object), the subdivision stops. On the other hand, if the cell is heterogeneous, that is, intersected by one or more of the object's bounding surfaces, the cell is subdivided further into eight subcells. The subdivision process stops when all the leaf cells are homogeneous to some degree of precision. The advantage of using an octree representation is that any arbitrarily shaped object, can be represented to the precision of the smallest cell.

All the 3D points recovered from the first image pair are inserted directly into the octree. All other 3D points that are recovered from later pairs of images are rotated about y axis by θ, where θ is the rotational angle of the left image, and then inserted into the octree. As mentioned before, θ can be directly read from the rotational stage.

Chapter 5

Experimental Results and Error Analysis

5.1 Experimental results

Some experimental results with the shape from rotation using stereo algorithm are presented in the following figures. A simple calibration object was used to calibrate the camera system and rotating platform as discussed earlier. Each image was digitized to 128x128x8 bits.

An existing stereopsis algorithm [18] was used to determine the disparities estimated from successive frames and their uncertainty. Figure 5.1 shows the first frame and second frame from the rotational sequence of a coffee cup, and the disparity and certainty results of the images pair after applying stereo algorithm. In order to display the results of disparity and certainty, which are represented in floating point, a converting program has been applied to map all disparities or certainties into all intensity between 0 to 255. In other words, the minimum disparity point will have intensity value of 0 (black), the maximum disparity point will have an intensity value of 255 (white). Note, that the disparity and certainty values of all background points are equal to zero. The cup was rotated 5^o between measurements. The front and top views of recovered cup are given in Figure 5.2. The recovered 3D object viewed from 16 different orientations and the same reconstructed object sliced into layers are given in Figure 5.3. Note that poorly textured regions give fewer responses than highly textured regions. This results from a limitation of the stereo algorithm which is unable to obtain reliable disparity estimates for regions which lack good texture. The general shape of the cup (and the rotating platform) has been recovered. The hollow nature of the cup, and the handle are clearly visible.

Figure 5.4 shows the first and second frame from the rotational sequence of a toy, and the disparity and certainty results of the pair of images after applying the stereo

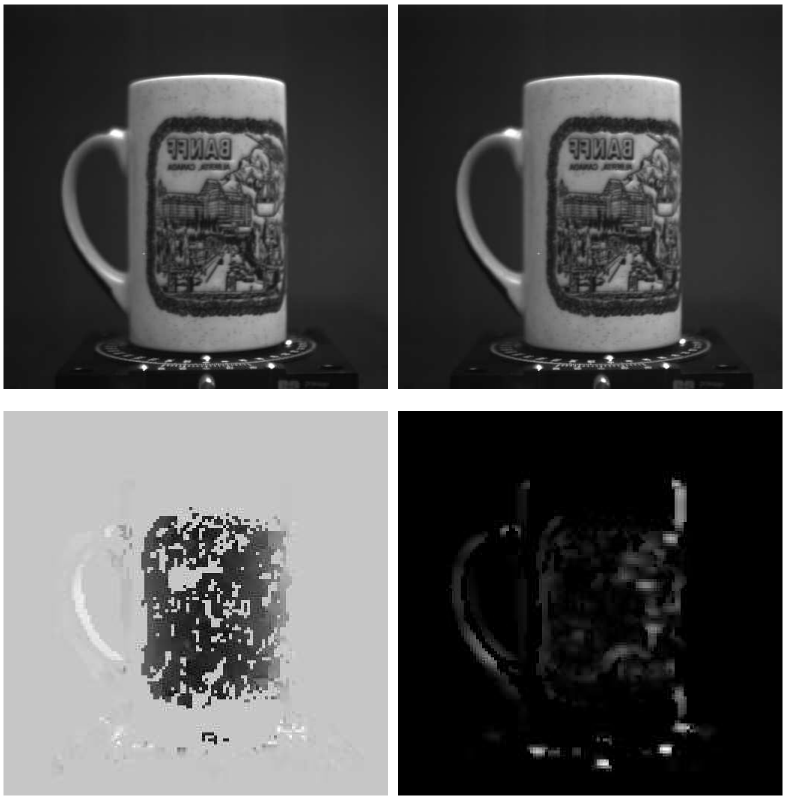

Figure 5.1: Experimental results of recovering a cup (a)first image in sequence (b)second image in sequence (c)disparities of the first two images (d)certainty of the recovered disparities

(a) Front view of the recovered 3D surface of the cup

(b) Top view of the recovered 3D surface of the cup

Figure 5.2: Front and top views of the recovered cup.

(a) 16 different orientation views of the recovered cup

(b) views of different sliced layers of the recovered cup

Figure 5.3: Different orientation views of the recovered 3D surface of the cup

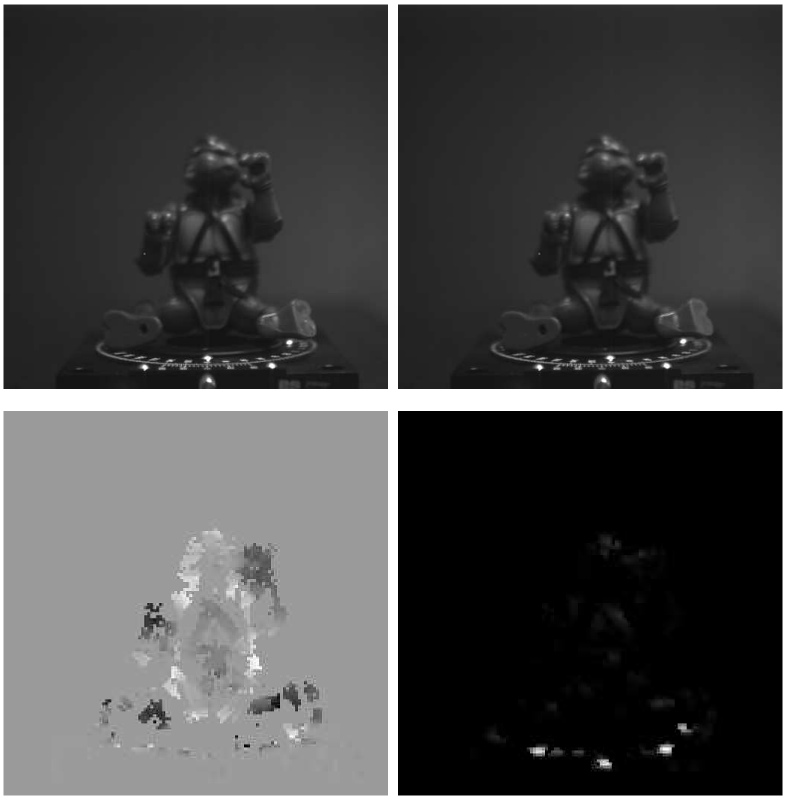

Figure 5.4: Experimental results of recovering a toy (a)first image in sequence (b)second image in sequence (c)disparities of the first two images (d)certainty of the recovered disparities

(a) Front view of the recovered 3D surface of the toy

(b) side view of the recovered 3D surface of the toy

Figure 5.5: Front and side views of the recovered toy.

(a) 16 different orientation views of the recovered toy

(b) views of different sliced layers of the recovered toy

Figure 5.6: Different orientation views of the recovered 3D surface of the toy

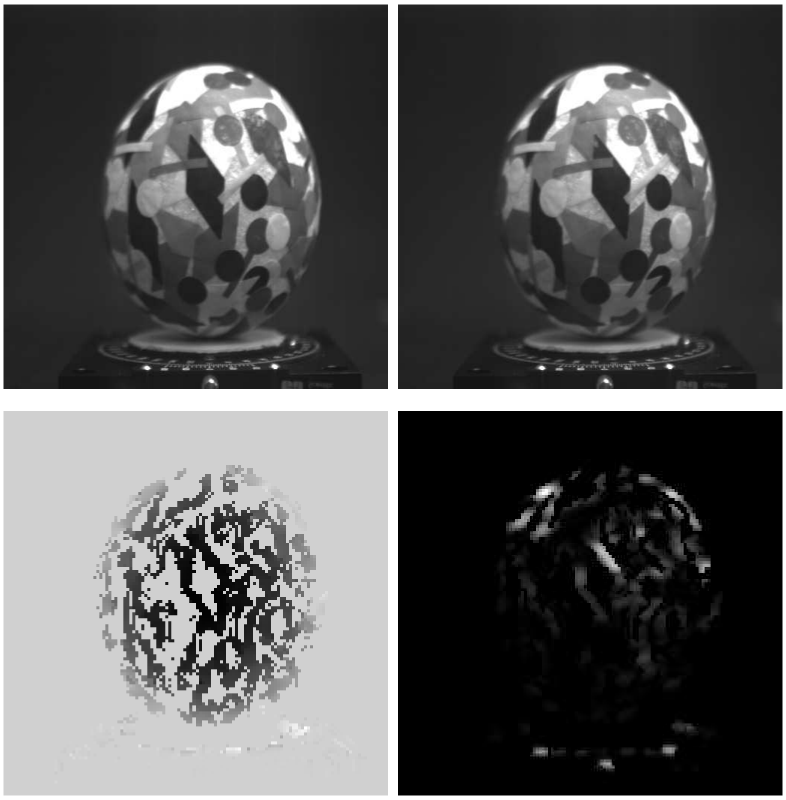

Figure 5.7: Experimental results of recovering a sphere (a)first image in sequence (b)second image in sequence (c)disparities of the first two images (d)certainty of the recovered disparities

algorithm. The toy was rotated 5^o between measurements. The front and the side views of recovered toy are given in Figure 5.5. The recovered 3D object viewed from 16 different orientations and the same reconstructed object sliced into layers are given in Figure 5.6. The 3D structure is clearly visible.

Figure 5.7 shows the first and second frame from the rotational sequence of a spherical object, and the disparity and certainty results of the image pair after applying the stereo algorithm. Note that in this figure, the real image of the sphere have been stretched vertically due to different aspect ratios of the camera and output device. This shearing error is one of the issues addressed through the calibration process mentioned in the previous chapter. The sphere was rotated 5^o between measurements. The front and top-front views of recovered sphere are showed in Figure 5.8. The recovered 3D object viewed from 16 different orientations and the same reconstructed object sliced into layers are given in Figure 5.9. The 3D structure is clearly visible and spherical.

(a) Front view of the recovered 3D surface of the sphere

(b) Front-top view of the recovered 3D surface of the sphere

Figure 5.8: Front and front-top views of the recovered sphere.

(a) 16 different orientation views of the recovered sphere

(b) views of different sliced layers of the recovered sphere

Figure 5.9: Different orientation views of the recovered 3D surface of the sphere

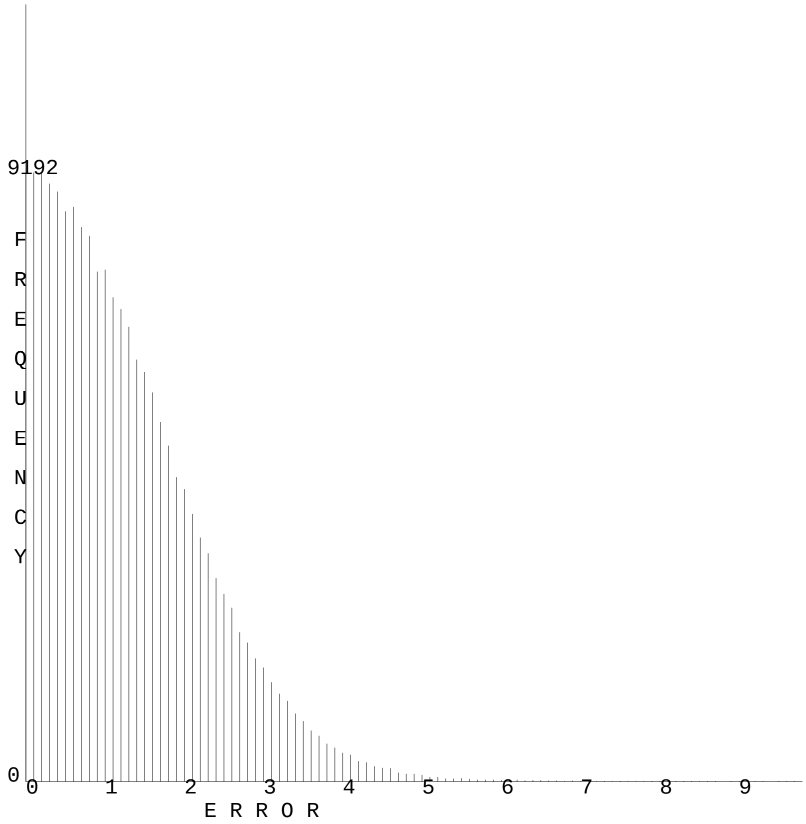

Figure 5.10: Histogram of the absolute errors (in mm unit)

5.2 Error Analysis

To examine the accuracy of the 3D recovered surfaces obtained with our technique, a known spherical object was used. The shape from rotation using stereo algorithm was applied to the spherical object in order to obtain a recovered surface model of the object. Given the true radius of the spherical object, (r_m), and the center of the sphere (a, b, c), then the error of a particular recovered 3D point (x_i, y_i, z_i), err_i, can be calculated as $err_i = \left| r_m - \sqrt{(x_i - a)^2 + (y_i - b)^2 + (z_i - c)^2} \right|$. For all recovered 3D surface points, the average error was calculated as:

$$err = \sqrt{(\sum_{i=1}^{n} (err_i)^2)/n} \tag{5.1}$$

A spherical object of radius 50mm was used to compute the error in the recovered object (the same object that appear in Figures 5.7 - 5.9). The root mean squared error was 1.7mm, which is about 3.3% of the sphere's radius. This is a relatively accurate result although the spherical object is not completely spherical. Thus the root mean square error probably overestimates the true error in the recovered object. As one can see from the histogram of absolute errors in Figure 5.10, most of the errors are less than 2mm, or about 4.0% of the radius. Figure 5.11 shows the error distribution over the

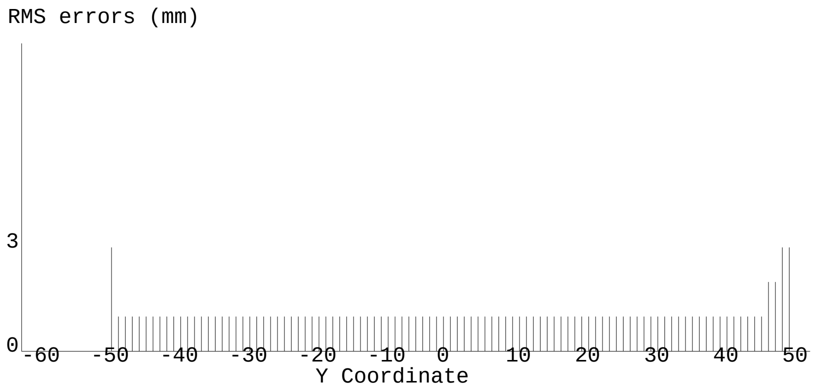

Figure 5.11: Error distribution (in RMS)

y coordinate of the sphere. When a recovered point is located near the top or bottom of the spherical object, its error increases dramatically. This is because the stereopsis algorithm generates greater errors due to the fact that the tangent of the recovered point on the sphere is changing rapidly near the top or bottom of the sphere.

Chapter 6

Discussion and Future Work

The techniques described in this thesis perform 3D shape reconstruction with rather simple equipment. The calibration system is rather simple, and no special light sources are required. The recovered 3D surfaces are quite accurate. Furthermore, since a general stereo algorithm and octree are used to reconstruct and represent the object, the object can be of arbitrary shape.

Shape from rotation using stereo is a very practical approach for building 3-D models of an object. The goal of this work was to produce a locally accurate model of shape and intensity with relatively simple equipment and non-restricted models. As such, it should be useful in a variety of robotics and computer vision tasks, as well as providing novel input for objects to be used in computer animation systems.

Some tasks remain. A process is required to trim isolated spurious responses from the object description and some mechanism should be used to merge the recovered data points into a surface or more compact volumetric model of the object. Furthermore, integrating the data into a CAD package would also be an interesting research topic.

Appendix A

Terminology

- **Bandwidth:** The difference or ratio between the limiting frequencies of a continuous frequency spectrum.

- **Irradiance:** Irradiance is the power per unit area ($W \cdot m^{-2}$–watts per square meter) of radiant energy falling on a surface.

- **Convolution:** Convolution is a very important image-processing operation, and is a basic operation of linear system theory. The convolution of two functions f and g is a function h of a displacement y defined as

$$h(y) = f * g = \int_{-\infty}^{\infty} f(x)g(y-x)dx$$

- **Homogeneous Coordinates:** Homogeneous Coordinates are redundant: a point in Cartesian n-space is represented by a line in homogeneous $(n+1)$-space. Thus each (unique) Cartesian coordinate point corresponds to infinitely many homogeneous coordinates. For example, a point (x, y, z) in Cartesian Coordinates can be represented as (wx, wy, wz, w) in Homogeneous Coordinates.

- **Gaussian (G):**

$$G = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x^2+y^2)/2\sigma^2},$$

  σ is the standard deviation of Gaussian (also defined as the scale of the filter). The Gaussian is a low pass filter.

- **Gabor ($Gabor$):**

$$Gabor(x.\sigma, k_0) = e^{ixk_0} G(x; \sigma),$$

  where $G(x;\sigma)$ is a Gaussian window with standard deviation σ, and k_0 is referred to as the peak tuning frequency (in radians) to which the filter responds maximally.

Bibliography

[1] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1982.

[2] A. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 11:11–30, 1981.

[3] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In A. R. Hanson and E. M. Riseman, editors, *Computer Vision Systems*, pages 3–26. Academic Press, New York, New York, 1978.

[4] R. A. Brooks, R. Greiner, and T. O. Binford. The acronym model-based vision system. In *Sixth International Joint Conference on Artificial Intelligence*, pages 105–113, Tokyo, Japan, 1979.

[5] J. F. Canny. Computational approach to edge detection. In M. A. Fischler and O. Firschein, editors, *Readings in Computer Vision*, pages 591–604. Morgan Kaufmann Publishers Inc., New York, 1987.

[6] I. Carlbom, I. Chakravarty, and D. Vanderschel. A hierarchical data structure for representing the spatial decomposition of 3-d objects. *IEEE Computer Graphics and Applications*, pages 24–31, 1985.

[7] H. H. Chen and T. S. Huang. A survey of construction and manipulation of octrees. *CVGIP*, 43:409–431, 1988.

[8] S. J. Dickinson, A. P. Pentland, and A. Rosenfeld. From volumes to views: An approach to 3-d object recognition. *CVGIP: Image Understanding*, 55(2):130–154, 1992.

[9] D. J. Fleet, A. D. Jepson, and M. R. M. Jenkin. Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210, 1991.

[10] D. Gabor. Theory of communication. *J. Inst. Electr. Eng.*, 93:429–457, 1946.

[11] S. Ganapathy. Decomposition of transformation matrices for robot vision. In *IEEE International Conference on Robotics*, pages 130–139, Atlanta, 1984.

[12] W. E. L. Grimson. *From Image to Surfaces (A Computational Study of the Human Early Visual System).* The MIT Press, Cambridge, Massachusetts, 1981.

[13] B. K. P. Horn. Non-correlation methods for stereo matching. *Photogramm. Eng. Remote Sens.*, 49(5):535–536, 1983.

[14] B. K. P. Horn. *Robot Vision.* The MIT Press, Cambridge, Massachusetts, 1986.

[15] C. J. Jackins and S. L. Tanimoto. Oct-trees and their use in representing three-dimensional objects. *CVGIP*, 14:249–270, 1980.

[16] M. Jenkin and A. D. Jepson. The measurement of binocular disparity. In Z. Pylyshyn, editor, *Computational processes in human vision: An interdisciplinary perspective*, pages 69–98. Ablex Press, Norwood, NJ, 1988.

[17] M. R. M. Jenkin. *Visual Stereoscopic Computation.* PhD thesis, Department of Computer Science, University of Toronto, 1988.

[18] M. R. M. Jenkin and A. D. Jepson. Local surface structure from local phase measurements. *Submitted for publication*, 1992.

[19] M. R. M. Jenkin, A. D. Jepson, and J. K. Tsotsos. Techniques for disparity measurement. *CVGIP: Image Understanding*, 53(1):14–30, 1991.

[20] K. Langley, T. Atherton, R. Wilson, and M. Larcombe. Vertical and horizontal disparities from phase. In *Proc. ECCV 90,Antibes*, volume 427, pages 315–325, Berlin/New York, 1990. Springer-Verlag.

[21] D. Marr. A note on the computation of binocular disparity in a symbolic, lowlevel visual processor. Technical Report Memo 327, MIT Artificial Intelligence Laboratory, 1974.

[22] D. Marr. *Vision.* W. H. Freeman and Company, New York, New York, 1982.

[23] D. Marr and E. C. Hildreth. Theory of edge detection. *Proc. R. Soc. Lond. B*, 207:187–217, 1980.

[24] D. Marr and T. Poggio. A theory of human stereo vision. *Proc. R. Soc. Lond. B*, 203:301–328, 1979.

[25] J. Mayhew, J. Frisby, and P. Gale. Psychophysical and computational studies towards a theory of human stereopsis. *Artificial Intelligence*, 17:349–385, 1981.

[26] D. Meagher. Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer. Technical Report IPL-TR-80-111, Image Processing Laboratory, Rensselaer Polytechnic Institute, October 1980.

[27] D. Meagher. Geometric modeling using octree encoding. *CVGIP*, 19:129–147, 1982.

[28] A. Papoulis. *Probability, Random Variables, and Stochastic Process.* McGraw-Hill, Singapore, 1991.

[29] A. P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–331, 1986.

[30] H. Samet. *Applications of Spatial Data Structures (Computer Graphics, Image Processing, and GIS).* Addison-Wesley, Reading, Massachusetts, 1990.

[31] T. D. Sanger. Stereo disparity computation using gabor filters. *Biol. Cybernet.*, 59:405–418, 1988.

[32] Y. Shirai. *Three-Dimensional Computer Vision.* Springer-Verlag, Springer-Verlag, Berlin, New York, 1987.

[33] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-Level Vision.* Kluwer Academic Publishers, Boston, Massachusetts, 1989.

[34] R. Szeliski. Real-time octree generation from rotating objects. Technical Report 90/12, Digital Equipment Corporation,Cambridge Research Lab, December 1990.

[35] R. Szeliski. Shape from rotation. Technical Report 90/13, Digital Equipment Corporation,Cambridge Research Lab, December 1990.

[36] R. Szeliski. Shape from rotation. In *Proceeding CVPR'91*, pages 625–630, Lahaina, Maui, Hawaii, 1991.

[37] H. Takahashi and F. Tomita. Self-calibration of stereo cameras. In *Proc. Second International Conference on Computer Vision*, pages 123–128, Tampa, Florida, USA, 1988.

[38] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *CVGIP*, 24:52–96, 1983.

[39] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models and 3d object reconstruction. *International Journal of Computer Vision*, 1(3):211–221, 1987.

[40] G. B. Whitham. *Linear and Nonlinear Waves.* Wiley, New York, 1974.