Reconstructing and Interpreting the 3D Shape of Moving Objects

F.P. Ferrie
B.Eng, M.Eng

Computer Vision and Robotics Laboratory
Department of Electrical Engineering
McGill University
Montréal, Québec, Canada

A thesis submitted to the Faculty of Graduate Studies
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

March 1986

Reconstructing and Interpreting the 3D Shape of Moving Objects

F.P. Ferrie
B.Eng, M.Eng

Abstract

This thesis investigates the problem of recovering the three-dimensional shape of an object from a sequence of views of that object. Shape is defined by a hierarchical representation in which an object is described by a set of surfaces and surface descriptors at the lower level, and as an abstraction of these surfaces in terms of volumetric primitives at the higher level. Objects in this domain are piecewise smooth in surface continuity, reflectance, and assumed to be temporally rigid. The purpose of addressing this problem is two-fold: (1) To gain insight into the more general nature of the vision problem of which this problem is a microcosm, and (2) To develop computational models which can be used to solve applied problems in computer vision.

The result of this effort is a framework for the shape problem and its implementation on a digital computer. This framework is composed of a representation, computational model, and a set of constraints by which a sequence of shaded images is turned into a three-dimensional composite geometric model made up of ellipsoids and cylinders. The computational process consists of 3 steps: (1) computing descriptions of the visible surfaces of an object in each view from a sequence of two-dimensional images; (2) computing inter-frame transformations describing object motion between successive views, thereby allowing for the reconstruction of surface descriptions in a world coordinate frame; (3) computing a composite geometric model from the reconstructed surface description.

In order to implement this framework, a number of problems fundamental to computer vision are addressed. These include the localization of surfaces in images, determining the correspondence of surfaces, and the inference of local geometric structure from surface differential geometry. The results of experiments are presented which illustrate the performance of the framework on real and artificial images, as well help clarify the concepts discussed.

Résumé

Cette thèse traite du problème de la reconstruction de la forme tridimensionnelle d'un objet à partir d'une séquence de vues de cet objet. La forme se définit par une représentation hiérarchique dans laquelle un objet est décrit par une ensemble de surfaces et de descripteurs de surface au plus bas niveau, et par une abstraction de ces surfaces en termes de primitifs volumétriques au niveau le plus élevé. Les objets de ce domaine sont lisses par parties en continuité et en réflectance, et sont par hypothèse temporellement rigides. Le but de s'adresser à ce problème est double: (1) mieux comprendre le problème de la vision en général, dont ce problème est un microcosme, et (2) développer des modèles opératoires utilisables dans la solution de problèmes appliqués de vision par ordinateur.

Le résultat de cet effort est un cadre de travail pour le problème de la forme et sa réalisation sur un ordinateur digital. Ce cadre de travail est composée d'une représentation, d'un modèle opératoire, et d'un ensemble de contraintes par lesquelles une séquence d'images ombrées est transformée en un modèle géométrique tridimensionnel composé d'ellipsoïdes et de cylindres. Il y a 3 étapes dans le processus opératoire: (1) le calcul de descriptions des surfaces visibles d'un objet à partir de chacune des images planes de la séquence; (2) le calcul des transformations décrivant le mouvement de l'objet d'une image à la suivante, rendant possible la reconstruction de descriptions des surfaces dans un système de coordonnées universel; (3) le calcul d'un modèle géometrique composé à partir de la description des surfaces.

Afin de mettre en oeuvre ce cadre de travail, un certain nombre de problèmes fondamentaux à la vision par ordinateur sont traités. Il y a la localisation des surfaces dans une image, la détermination de la correspondence entre surfaces, et l'inférence d'une structure géométrique locale à partir de la géométrie différentielle d'une surface. Les résultats d'expériences présentés illustrent la performance de ce cadre de travail sur des images réelles et artificielles, et aident à clarifier les concepts discutés.

Acknowledgements

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant A4156, by an FCAC Grant awarded by the Department of Education, Province of Québec, under Grant EQ-633, and the Medical Research Council of Canada under Grant MRC-3236.

I would first of all like to thank my supervisor Dr. Martin Levine for his support of my work through two degree programs, and for giving me my introduction to the field. Many individuals at the laboratory also helped shape my thinking, and to them I will abways be grateful. To Dr. Steven Zucker for motivation in looking beyond the obvious and criticism during the course of this research, to Yvan Leclerc and Norah Link for their feedback and encouragement, and to the many others who have passed through the lab over the years for their ideas and enthusiasm, thank you. I would would also like to thank Dr. Peter Noble of the Faculty of Dentistry for his patience as I attempted to understand some of the basics of cell biology, and Dr. M.M. Frojmovic of the Department of Physiology for preparing the data which I used in my experiments. Thank you to Paul Freedman and Pierre Parent for help in the preparation of this document, and especially to John Lloyd for lending his artistic talents.

A lot of friends shared this journey. Without their help and support it is doubtful whether the work would ever have gotten started, never mind completed. Warmest regards to Steve Zucker, Yvan Leclerc, Norah Link, Peter Sander, Pierre Parent, Harold Hubschman, Kamal Gupta, and the rest of the Marauders near and far for the cameraderie that made the whole experience worthwhile. Norah Link and I shared a friendship as well as an office, and Mike Brassard was always there to pick up my spirits. To you both, thank you. Finally, I would like to thank my family for their constant encouragement, and last and most of all to Rosemary for her love and patience.

March 1986

Contents

List of Figures

Chapter 1 Introduction

1.1 Problem Description

How can the three-dimensional shape of a solid object be recovered from a sequence of views of that object? The answer is, of course, a summary of much of the task of vision. During the past thirty years a lot of progress has been made in solving various components of the vision problem, but these have not mapped into the "seeing machines" that were thought to be close at hand in the late sixties. In fact, this apparent failure has caused some researchers to re-evaluate their earlier positions on frameworks for solving the vision problem [Witkin & Tenenbaum 1984]. A widely held belief is that the task of vision proceeds from lower to higher levels of representation, namely from images to surfaces to objects [Gibson 1950, Marr 1982, Barrow & Tenenbaum 1978]. But what is not clear is how this transition in representations is accomplished. This brings up a number of important questions which are the central focus of this thesis:

- What is the nature of these different levels of representation? In particular, what is the essential structure that must be captured in each?
- How are these different descriptions computed and what are the constraints that make the computations possible?

- How much "top-down" feedback is necessary in computing the different descriptions? In what form is it represented explicitly and how is it manifest implicitly in the form of assumptions about *what* is being computed?

We define a *framework* for vision as a set of three components: (1) a representation for a class of objects in a specified domain, (2) a computational model for computing a description from sensory data, and (3) a set of constraints that provide the necessary boundary conditions for what are usually underdetermined problems. Given a particular framework for vision, then, the above questions can be summarized by asking: how does one determine the competence of a given framework?

The complete answers to these questions are unlikely to be found in the near future due to the complexity of the vision problem in the general domain. However, there is still much that can be learned by studying the problem in a restricted domain where the necessary constraints exist by which a particular problem may be solved (see also Binford [1982]). By doing so, a more general insight can be provided into fundamental problems, much in the same way that experimental physics serves to complement the work of theoretical. In an attempt to answer the questions posed earlier, we will develop a framework for vision in a domain of piecewise smooth, solid objects [1]. This will provide the necessary constraints by which we can derive a computational model for transforming images, such as the one shown in figure 1a, into a geometric description composed of ellipsoids and cylinders (figure 1b). Analyzing this model, on the other hand, is a much more difficult task. While local analysis is possible, the effects of concatenating successive results (in this case making the transitions between different levels of representation) are less predictable because of the non-linear nature of the model. For this reason the model is supplemented by an implementation which allows for actual experiments. The model and its implementation provide a basis for answering the above questions in a

---

[1] Similar to the Playdough World of Barrow & Tenenbaum [1978].

specific domain. For in order to arrive at a working model that solves real problems we will have to find answers for each of these questions in a specified domain.

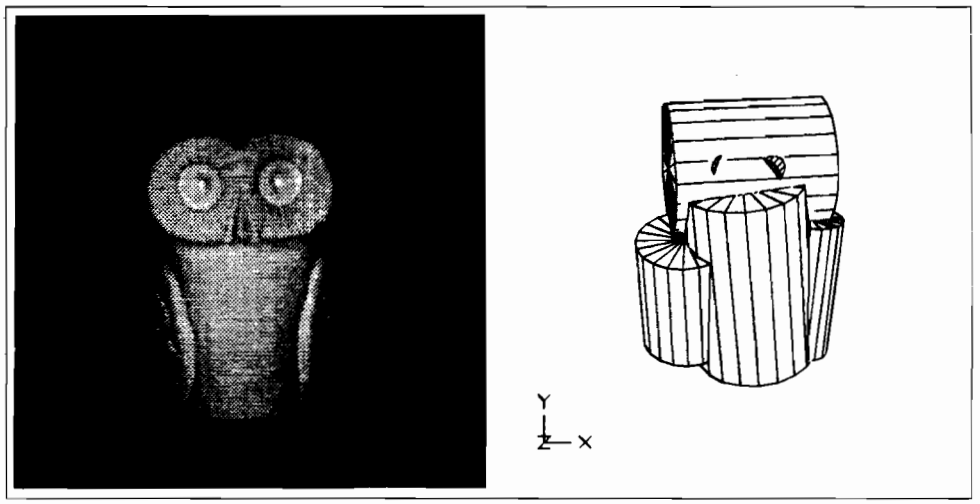

Figure 1a & 1b (a) Image of a stone owl sculpture (b) Ellipsoid-cylinder model computed from the image

1.2 Approach

The key to our approach is an organization of data that makes explicit salient features of the structure of an object at two fundamental levels of abstraction. An object is represented as a surface at one level and as the volume occupied by a conjunction of ellipsoids and cylinders at the other. The transformation from the surface to the volumetric description is accomplished by exploiting a constraint from differential geometry applicable in our domain: if a surface $S(t)$[2] is decomposed into a set of non-overlapping patches, $S_i(t)$, where each $S_i(t)$ is homogeneous in normal curvature [do Carmo 1976], then the geometric object of which $S_i(t)$ is a part may be inferred on the basis of local properties. We refer to this process as geometric inference. Now if objects in the domain of consideration are assumed to admit a

---

[2] The index t refers to an observation of a surface S at a particular time t. The object whose surfaces are being viewed can either move about the observer, or vice-versa.

decomposition into volumetric primitives, then it is likely that this can be inferred from the surface decomposition.

The surface decomposition is also useful from another point of view. It is assumed that the description of an object is computed by observing a sequence of views, as reflected in the time dependency of $S(t)$ and $S_i(t)$. Unless the object is familiar to the viewer or possesses a degree of symmetry, it is unlikely that a complete description can be computed from a single view. Otherwise, for example, how can information about the "back" of an object be acquired? For this reason we must consider how information is integrated from different views. One way of doing this is to locate corresponding structure in different views and then try to determine how the object (or viewer) moved in order to account for its different appearance. If the object is assumed to be temporally rigid, then geometric constraints can be exploited such that this motion may be computed from the coordinates of corresponding structures. The task of locating these structures is referred to as the correspondence problem and its solution depends on localizing matching structure across different views of the object. It is made easier by attempting to match the structure that is intrinsic to the object, the same structure that happens to be made explicit in the decomposition of $S(t)$.

We can take the process of decomposition one step further and decompose each $S_i(t)$ into a set of intrinsic feature vectors $F_j(t)$. This allows for a more local definition of structure which is necessary in order to simplify the matching process as well as to provide a means of coping with problems caused by occlusion. By having a proper notion of local structure, we are able to formulate the matching problem precisely. This will not only allow us to compute the interframe transform T_{ab} which maps $S(a)$ into $S(b)$, but also to determine the existence of T_{ab}. Without such a test there would be no way to determine when correspondence is lost, as when one object occludes another. We can exploit this information as a signal and to try to re-establish correspondence at a more global level. By solving the correspondence problem through the entire sequence of views, a complete description of the surfaces of the object can be computed. It is at this point that constraints about the differential geometry

of the resulting three-dimensional surfaces can be used to make inferences about the object's local geometric structure.

The representation for shape that we have just outlined is summarized in figure 2. By abstracting it in the manner shown, we are able to formulate the computational steps involved as three distinct sub-problems:

1. The location[3] of surfaces from different viewpoints and the computation of their intrinsic properties as defined in terms of differential geometry.
2. The estimation of object (or viewer) motion and the reconstruction of surfaces in a world coordinate system.
3. The computation of a composite geometric model and its subsequent interpretation from the above surfaces and their intrinsic features.

Each of the above sub-problems has received attention by a number of researchers (c.f. section 2.1), but these problems have usually been studied independently of each other. The advantage of maintaining a global perspective, however, is that representations and computational models can be designed to take advantage of mutual constraint information. While such constraints are often not apparent when considering problems in isolation, they become so when considering how one level of representation is transformed into another. As we shall see through the course of this thesis, representations are usually a compromise between what is desirable and what is computable [Marr 1982]. Even so, if it were possible to compute any desired feature, it is not always clear *what* to compute. We believe, however, that by considering how an object is represented at different levels of abstraction and how transitions are made between the various levels, it becomes more obvious what the essential structure is that has to be captured at a particular level.

---

[3] By the word "location" we mean the task of determining which regions in a view correspond to the surfaces of an object, followed by the estimation of the relative depth of the visible surfaces as seen from a particular viewpoint.

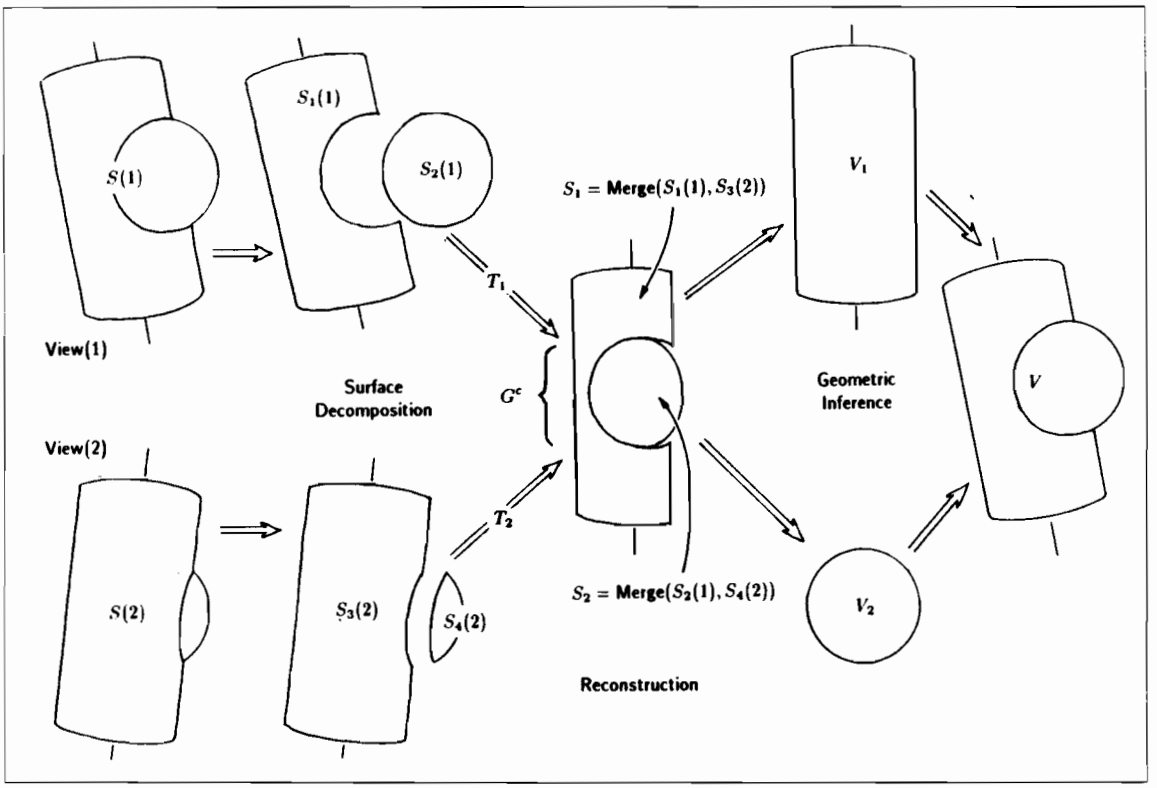

Figure 2 Components of a representation for shape

By examining these transitions in detail, one can arrive at constraints that directly influence what is being computed. For example, if surfaces are considered to be important for representing objects, then one must consider how surface cues are encoded in images. The constraints imposed by the process of image formation specify what needs to be computed in order to extract surface information. Similarly, in inferring the local geometric structure of surfaces, constraints from differential geometry tell us the necessary features that must be computed in order to make inferences. Because of the largely underdetermined nature of the vision problem, we require constraint information to both make possible and simplify the processes by which various features are computed [Zucker 1981]. As this work will serve to demonstrate, the success of a framework for solving a particular problem is largely related to the validity of the constraints applied to its solution.

Constraint knowledge exists in many forms. It may be generic in the form of intrinsic properties that are common to all physical objects or it may be particular to a specific domain.

Consequently, in order to attain the utmost generality, we wish to exploit generic constraints as much as is possible. The question is to what extent is this possible. Invariably, it is not always possible to compute a unique description of an object, as anyone who has ever viewed an unfamiliar object will attest. A particularly difficult problem is that of determining how much domain-dependent knowledge is required for a specific vision task. These tasks are often viewed conceptually in a hierarchical context, where order reflects a task's dependency on the domain. In this context, lower level tasks tend to be more general in the nature of their application, whereas higher level tasks are more specific. We refer to domain-dependent knowledge as "top-down" and generic constraints as "bottom-up", reflecting this hierarchical context. One method of determining domain-dependent knowledge requirements is to proceed with a "bottom-up" approach as far as is possible until additional constraints are required. However, this may still not take into account constraints that are implicit in the domain itself. Problems such as these are particularly difficult to analyze independent of applications. This is another reason why it is important to study applied problems in computer vision.

1.3 Motivation

The principal motivation behind our work is an attempt to answer the questions posed in Section 1.1. We will do so by building a framework for the shape problem that is defined in terms of the sub-problems outlined in Section 1.2. By implementing this framework in a computer vision system, we have a means of accessing its competence in solving problems that are fundamental to computing and representing 3D shape. There are many subtle factors involved that usually escape inclusion in computational models, largely due to reasons of complexity as well as subtlety. By considering specific applications, however, we can take advantage of constraints that help minimize this complexity. Furthermore, experiments can be performed by which computational models are refined until their competence is sufficient to solve a particular vision task. The most important results of this process are a better understanding of the vision problem, and progress towards systems that are capable of solving

real problems.

Much emphasis has been placed in recent years on the interpretation of visual shape. However, in order to navigate and function in an environment it is also necessary to consider how to compute descriptions that permit structural descriptions and measurements of shape[4]. Another motivation for this research is the investigation of how the above descriptions are computed, in particular how differential features can be exploited in this process. If one takes the Gibsonian view that form is computed from perceived surfaces, what is it then, about their structure that gives rise to higher level abstractions? How, for example, are models such as Generalized Cylinders [Binford 1971] computed from a representation based on surfaces? Towards this end we will develop a notion of "curved edges" corresponding to extrema in surface normal curvature that can be used to parse a surface $S(t)$ into a set of patches $S_i(t)$. These patches are assumed to lie on the surfaces of geometric primitives, the conjunction of which forms a composite model of an object in a scene. The primitives themselves are inferred from surface patches by applying constraints from differential geometry. These constraints essentially allow the selection of a particular model which is subsequently parameterized by fitting it to data.

A third motivation for this work concerns the fact that, unless there exists a particular symmetry, an object *must* be observed from different views. This is particularly true in cases where measurements of shape are required. To address this problem, we introduce a representation for multiple views called a *surface graph* and its realization as a set of *dynamic intrinsic images* (DII's). A DII is a classical intrinsic image in the sense of [Barrow & Tenenbaum 1978] in which differential features of a surface are made explicit. However, it differs in one key respect: it contains an associated transformation which maps viewer-centered coordinates into a common coordinate representation.

---

[4] For example, in order to grasp a box, a robot manipulator system is more interested in knowing the precise dimensions of the box than the simple fact that the object is a box.

In order to obtain this transformation, the set of inter-frame transformations mapping successive views into the coordinate systems of their successors must be computed. The major difficulty here is in accurately determining the correspondence between adjacent views. We will show how the differential features made explicit in DII's can also be exploited in the matching process. This gives rise to a view of the correspondence problem as a correlation of the structure of local neighbourhoods. Computation of the required transformations allows an implicit reconstruction of an object's surfaces in terms of a set of DII's. Once this is accomplished, geometric inference can be carried out over the set of reconstructed surfaces. The conjunction of the resulting primitives serves as a composite model of the object under view.

An important difference between our work and other approaches to surface reconstruction is in the role of correspondence. The conventional structure from motion paradigm [Ullman 1979] exploits the geometric relationship between corresponding sets of projected surface features to recover their 3D locations. This assumes that such features can be localized without first recovering surfaces, and that correspondence exists between two views. Such is not the case in our problem domain where surfaces tend to be smooth and object motion sometimes unpredictable. In fact, we require detailed structural information about surfaces to determine whether correspondence exists in the first place.

Recent applications of computer shape analysis to cell biology (see for example [Noble & Levine 1986]), provide a fourth motivation for this research. This concerns the application of three-dimensional shape analysis to some specific problems involving analysis of blood platelet geometry [Frojmovic and Milton 1982] and quantification of the relationship between cell shape and locomotion [Youssef 1982, Noble & Levine 1986]. Virtually all of the previous research in these areas is based on the two-dimensional analysis of cell shape which is inherently ambiguous. The present effort calls for the reconstruction of cell surfaces by exploiting known parameters of the imaging process such that the resulting data may be interpreted as shaded surfaces. This is, at best, only an approximation of how images are actually formed by light

microscopy and the interesting question here is how well the resulting shape model will fit biological data under these assumptions.

1.4 Overview of the Thesis

This thesis is composed of four principal chapters. Chapter 2 begins with a background presentation covering pertinent research related to 3D shape representation and computation. It then introduces a framework for shape as applied to a domain of smooth, solid objects. The shape model is based on the premise that a representation for shape requires descriptions of salient characteristics of an object at different levels of abstraction. These range in scope from local to global and are rooted in the physical structure of an object as reflected in its surfaces. It is our view that the process of describing the shape of an object consists of computing these descriptions. In particular, a computational model is presented that consists of three sub-tasks: the description of the structure of surfaces, the integration of surface information from different views by computing the motion of an object, and the inference of the geometric structure implied by the set of reconstructed surfaces. Chapters 3 to 5 focus on each of the above individually, specifically those considerations which enable the computational model to work in practice.

Chapter 3 focuses on the first sub-problem described above, computing a description of the structure of surfaces. In the context of our model, this consists of computing the surface graph description as an equivalent set of intrinsic images from a single viewpoint. The basis for computation is a shaded image of a piecewise smooth surface. It is shown that in spite of the uncertainty in determining local surface orientation, reasonable estimates of surfaces can be obtained with appropriate constraints. The computation of differential structure is then effected by locally approximating the resulting surface with a suitable interpolation function. Principal curvatures and directions for each point on the surface are readily computed from the parameters obtained in the interpolation process. Curvature information is used to decompose

the surface into a set of component patches with similar generic properties, to identify the local geometric characteristics of each patch, and to describe each patch on a pointwise basis in terms of intrinsic feature vectors. These vectors define what we call the structure of the local neighbourhood of a surface and are at the heart of the correspondence process. The chapter also presents results from experiments to highlight and clarify this approach to surface representation.

In Chapter 4 we deal with the problem of determining the relationship between two views (i.e. the motion transformation), and how this information can be used to integrate information across them. This requires a solution to the correspondence problem in which distinguished surface features are matched across adjacent views. The chapter presents a solution to the problem in terms of matching the structure of local neighbourhoods, as well as a computational model for doing so. An important feature of this model is its ability to determine when correspondence between two views is uncertain or nonexistent. In these circumstances, the local correspondence model breaks down and must be supplemented with a more global approach; in our case, matching based on occluding contour. An analysis of the correspondence model, based on experimental data, is also presented which investigates the role of differential features in the matching problem. Once the set of interframe transformations is computed, the surfaces of an object, as well as the composite surface description, can be reconstructed by mapping each viewer-centered description into a common reference frame and eliminating overlap. Results are presented in which composite surface graphs are computed from motion sequences of both artificial and real data. We can determine, at least qualitatively, the validity of the resulting descriptions by turning them into polygon descriptions which can be conveniently displayed as shaded images.

Chapter 5 focuses on the *geometric inference problem*, which is defined as that of inferring the geometric structure of an object that is implied by its surfaces. We begin with the assumption that objects in the domain can be represented as conjunctions of primitive shapes such as ellipsoids and cylinders. Next, the problem is sub-divided further into three sub-

problems:

1. The inference of an appropriate primitive for a particular surface patch.
2. The parameterization of each primitive based on features of the surface patch and the piecing together of primitives into a global representation for the object.
3. The interpretation of the global representation as some object in the domain.

Because the surface graph is composed of a set of homogeneous surface patches, primitives may be instantiated for each patch on the basis of the signs of the principal curvatures (which are the same everywhere for a particular patch). A parametric description for each primitive is subsequently obtained by fitting the appropriate model to data. To facilitate interpretation, however, local parametric descriptions must be somehow be aggregated into a more global representation for the object. For this reason local descriptions based on ellipsoids and cylinders are hierarchically organized based on size along the lines suggested by Marr [1982]. Geometric primitives are organized as a graph with nodes containing parametric descriptions of each primitive and links describing the spatial relationship between adjacent nodes. Interpretation is accomplished by either attempting to match the resulting graph structure against a prototype model, or else abstracting the structure further in terms of descriptive features which can be used for syntactic recognition. Our work is concerned mainly with the computation of the composite description and does not address the interpretation problem in detail. The second part of the chapter describes the results of experiments in which ellipsoid-cylinder models are computed for three different objects.

Chapter 6 concludes the presentation by summarizing the basic concepts put forward in this thesis, the results obtained, and what we view as the major contributions of this work. We close with some final thoughts on future work for which the present effort lays the groundwork.

Chapter 2 Towards a Model for Shape

The shape problem that we have defined summarizes much of the task of vision. A good place to begin, then, is to review some of the earlier work that led up to our investigation of the problem. The purpose here is to underscore the important issues and motivate our approach, which is summarized in the latter portion of the chapter.

2.1 Background

Webster's New World Dictionary [Webster 1953] defines shape as *"The quality of a thing that depends on the relative position of all points comprising its outline or external surface."* A fundamental problem for vision research is to define this notion of *quality* and apply it to describing objects in three dimensions. Further complicating the task is the fact that *quality* is dependent on the context of the viewer, in particular the conceptual level at which the description is formed. For example, in describing the physical characteristics of an individual, descriptors range from global characteristics such as height and weight, to local, more refined properties such as skin texture and eye color. In spite of this difficulty, there appear to be certain intrinsic properties of shape that serve to decompose a complex description into an aggregate set of simpler ones. This viewpoint gives rise to a definition of shape as *a set of descriptions of salient physical attributes of an object at different levels of abstraction.* Any approach to building representations for shape must describe what these different levels are,

how they are computed with the retinal array as the basic source of information, and how they are used to build meaningful descriptions of objects in three dimensions.

2.1.1 Shape and Surfaces

In order to build such a model, the nature of object descriptions and the constraints involved in their computation must be well-understood. The complexity of the natural scene domain, however, makes such understanding extremely difficult to obtain in the best of circumstances. For this reason, much of the early research into shape description and interpretation has been restricted to domains which could facilitate this understanding. The earliest example of such research is the *Blocks World* domain [Roberts 1965; Clowes 1971; Huffman 1971,76; Mackworth 1973; and Waltz 1975], to name but a few. This domain formed much of the basis of research for well over a decade and provided considerable insight into the nature of problems in machine vision. It also demonstrated that in order to deal with more natural scene domains consisting of curved surfaces, emphasis would have to be placed on the more fundamental problem of how the visible surfaces of objects are computed from images.

This inspired a number of Shape-from-X approaches:

- Shape from shading: [Horn 1970,75,77; Ikeuchi & Horn 1981; Woodham 1979; Pentland 1982a,82b; Smith 1983a,83b].
- Shape from texture: [Gibson 1950; Zucker 1975; Stevens 1980; Kender 1980; Ikeuchi 1980]
- Shape from contour: [Stevens 1980,81; Witkin 1980]
- Shape from disparity (stereo): [Levine et al. 1973; Marr & Poggio 1977; Grimson 1981]
- Shape from motion (discrete): [Ullman 1979; Nagel 1981; Dreschler and Nagel 1982; Hoffman 1982; Webb and Aggarwal 1981,82]

- Shape from motion (continuous): [Clocksin 1980; Prazdny 1979,80; Hoffman 1980; Waxman and Ullman 1983, Waxman and Kwangyoen 1984]
- Shape from motion stereo: [Nevatia 1976; O'Brien and Jain 1984; Waxman and Sarvajit 1984]

The common factor in each of these approaches is an attempt to invert the process of image formation, which is inherently under-determined, by the application of suitable constraint information. Some examples of these constraints are continuity arguments motivated by the physical cohesiveness of matter, correlation between 3D structure and its projected image based on general position assumptions, and the consistency of physical structure across a sequence of views. Continuity constraints have been used to a large extent in shape from shading problems and in interpolation processes associated with stereo [Grimson 1981; Terzopoulos 1982,84], and motion [Hildreth 1983]. The correlation of 3D and projected structure has been exploited in texture [Zucker 1975; Kender 1980; Ikeuchi 1980] and contour approaches to surface recovery [Witkin 1980; Stevens 1980,81]. Consistency (or continuity) across multiple views is the basis for shape recovery through motion, stereo, and optical flow. For a further discussion of the evolution and role of constraints in vision, see [Zucker 1981].

"Shape-from-X" approaches, however, are but a partial answer to the problem of surface recovery for they do not individually provide complete enough descriptions for subsequent processing. Three further problems must be addressed: (1) Coping with uncertainty or lack of estimates; (2) Combining information from multiple surface cues; and (3) Locating the boundaries which delineate different surfaces. The first or these problems has been investigated by Grimson [1981] and Terzopoulos [1982,84] in dealing with the interpolation of depth information in the recovery of surfaces from stereo images. In addition, Terzopoulos has also dealt with the second problem by including constraints imposed by surface orientation estimates in the interpolation process. Such processes, however, also require the location of delineating boundaries which serve to mark the locations of discontinuities on a surface. This third prob-

lem has been addressed from two different viewpoints, by Zucker [1982], and his colleagues [Zucker & Parent 1982,84; Link & Zucker 1985a,b] in the interpretation of flow fields, and [Leclerc & Zucker 1984; Leclerc 1986] in the location of discontinuities in an intensity image.

Given this rich variety of visual information and constraints it would seem that a reasonable goal for both human and machine vision would be the recovery of surfaces. However, what do these surfaces tell us about the structure of the 3D visual world? Witkin and Tennenbaum [1984] argue that surface recovery might *in itself* not be an appropriate goal, but rather the by-product of a more general process that seeks to identify causal relationships between 3D structure and its identity as projected in an image. They view the role of early vision as that of imposing organization on the primitive structure contained in these identities. A key element in this computational process is the resolution of the causal links between the physical structure of an object and its identity contained in an image. This is one of the problems that we address, with the restriction that the links are those between the 3D structure of an object and the identity reflected in its *surfaces*.

This issue has received attention principally in the Generalized Cylinders concept proposed by Binford [1971] and later work done by him and his co-workers, [Agin and Binford 1976; Nevatia and Binford 1977]. This concept was also adopted by Marr and his co-workers [Marr 1976; Marr & Nishihara 1977; Marr & Viana 1980] as part of a general theory of vision and was extended to include a notion of description at a variety of scales. In the work by Binford et al., generalized cylinder models of data were extracted from a combination of range and intensity measurements. Structural cues such as of axes symmetry and boundary information were used to hypothesize model elements which were then verified by fits to laser range data. The research of Marr et al. was directed more towards the direct computation of generalized cylinder models from structural cues contained in an image. Later attempts at interpreting range data are reported in [Duda, Nitzan, & Barrett 1979; Fischler & Barrett 1980; and Bhanu 1984]. Each of these approaches attempts to segment range information into polyhedral regions on the basis of iconic transformations and/or clustering techniques.

The resulting segmentation forms the basis for later interpretation of the scene. Bhanu's research is particularly interesting in that he considers integrating range information across adjacent views for the purpose of a full 3D polyhedral reconstruction of the underlying object.

The common thread in the above examples is the notion of exploiting local structural cues in the inference of more global structure. A formalization of this concept is precisely the domain of differential geometry *in the large:* "If a continuous geometrical figure is known to have a certain property in the neighborhood of *every one* of its points, then it is possible, as a rule, to deduce certain facts relating to the total structure of the figure" [Hilbert & Cohn-Vossen 1952]. Such a formalism fits in well with the paradigm of inferring the shape of an object from the structure of its surfaces, e.g. Marr's 2.5D sketch [1982] and Barrow and Tenenbaum's intrinsic image concept [1978].

Ideas and concepts from differential geometry have been applied to problems in vision by many researchers. Smith [1979] proposed the use of an extended Gaussian sphere as a representation for computer vision, a further discussion of which can also be found in [Horn 1979] and [Grimson 1979]. Pentland in his Ph.D dissertation [1982a] made use of constraints from differential geometry in the estimation of surface orientation based on local measurements of image intensity. A further significant contribution of his research was to show how local surface characteristics (e.g. Gaussian curvature) could be derived from intensity data and used to facilitate the interpretation problem. Zucker [1981; Link and Zucker 1985a,b] has shown how such constraints play a central role in low level grouping processes and has developed an extensive computational theory around them. Of particular importance is the fact that these grouping processes make explicit the structure of flow patterns which gives rise to surfaces and provides important cues for interpretation.

2.1.2 Integration and Motion

Thus far the discussion has centered on the computation of shape from a single viewpoint.

In many circumstances this is sufficient for the unique interpretation of objects that are either known to the viewer, or else possess a degree of symmetry that can be exploited in determining overall shape. In cases where the global shape of an object cannot be inferred from a single view or in cases where more precise measurements of shape are required, information from multiple views must somehow be combined. The scrutiny of a fashion model or a piece of artwork, for example, is a case in point. If one views shape as a process of computing descriptions at different levels of abstraction, the extension to multiple views would require that individual descriptions be combined in some coherent manner. In order to accomplish this, the spatial relationship between different views must be computed or else inferred from some knowledge about the scene or object in view.

It is well-known that the relationship between two views of a rigid body can be expressed as a linear transformation involving translations and rotations in three dimensions (e.g. [Newman & Sproull 1979]). The determination of this transformation requires the location of corresponding sets of points from the two surfaces and the solution of simultaneous equations in the ideal case. However, objects may not necessarily be rigid and the determination of surfaces and correspondence is made quite difficult due to the presence of noise and occlusion. In spite of these difficulties, human observers seem to have little difficulty in determining the motion and structure of objects [Johansson 1973; Ullman 1979; Hoffman 1982]. In fact, this can be accomplished from the observation of projected 3D point configurations, even in the absence of apparent 3D structure in the individual views [Johansson 1973; Hoffman 1982]. Such considerations led to the investigation of the constraints by which both the structure and motion of an object could be inferred from 2D image flows. The basic result of this research has been the derivation of non-linear equations relating 3D surface geometry to 2D image correspondences.

Ullman [1979] investigated the case of orthographic projection and was able to prove the uniqueness of solution for three successive views of four non-coplanar points. He also considered the case of perspective projection in which the object was assumed to rotate in

a vertical axis parallel to the image plane, followed by translation. A solution was obtained for two views of five non-coplanar points, but its uniqueness was not proven. Nagel [1981, Nagel & Neumann 1981] developed a simplified solution to the case of perspective projection by eliminating translation components. This significantly eased the computational burden, but the uniqueness of the solution was not determined. Tsai and Huang [1981] developed a set of nonlinear equations relating object motion to image point correspondences for small rotation angles. Fang and Huang later proved that these equations provided unique solutions for two views of nine points constrained not to lie on a second order surface passing through the viewing position [1983a]. Experimental analysis was later conducted in [Fang & Huang 1983b,84] that showed reasonable estimates could be obtained in practical situations for small rotation angles, and where relative translations were suitably restricted. In later work [Tsai & Huang 1984] obtained a general solution to the problem of perspective projection that proved unique for two views of seven points. These were constrained either to *not* be traversed by two planes with one containing the origin or to lie on a cone containing the origin. Furthermore, a linear solution was obtained that required only eight points in two views for a unique solution.

In spite of Ullman's structure from motion theorem [1979] or any other computational requirement for uniqueness, Johannson's experiments [1973] showed that humans could obtain structure from less than or equal to two visible points on each rigid object. This motivated others to seek additional constraints on the interpretation of visual motion. Webb and Aggarwal [1981] introduced the assumption that the motion of any object consists of translations and rotations about an axis that is fixed in direction over short periods of time. In effect this constrained the movement of points to circular trajectories which could be back-projected from the view plane allowing surface recovery. Hoffman [1982] introduced a planarity assumption in which points were constrained to translation and rotation in a plane. He was able to show that surface structure could be recovered uniquely from either three views of two points or two views of three points forming the endpoints of two rigidly connected rods.

Another approach to the problem of recovering structure and motion from 2D image cor-

respondences involves the continuous analogue to the discrete processes thus far considered, that is optical flow [Gibson 1950]. Prazdny [1979,80] was able to relate 3D surface structure to retinal flow velocity in a manner similar to Nagel's discrete approach. The result was a non-linear, third degree equation in three unknowns that had to be solved in iterative fashion. As with similar approaches in the discrete case, Prazdny's method did not ensure the uniqueness of solution. Hoffman [1980] presented an approach to computing shape from optical flow which was similar to the above except that the projection was orthographic. Thus far, optical flow analysis had been restricted to the information available from velocity vectors at each point in the field. A new approach reported in [Waxman & Ullman 1983] considered an analysis of flow fields in terms of a representation borrowed from fluid mechanics. This representation, the velocity-gradient tensor, describes the local deformation in a continuous flow field. The authors in turn augmented this description with an additional six independent gradients and related it to 3D surface structure through perspective projection. The result was a set of twelve non-linear equations in eleven unknowns that related measurements of functions of the flow field to parameters describing the structure and motion of the surface. The advantage of this approach, in spite of uniqueness problems due to the over-determined nature of the system, is that it appears to be tolerant of perturbation errors, a shortcoming with other methods.

2.1.3 Summary

The point of this brief overview has been to show that the three components of the shape problem defined earlier in section 1.2 have each been studied independently by a number of investigators. However, in order to solve this shape problem, it will be necessary to consider the relationships among the three in determining the shape of an object. What is important are those constraints that lead to a consistent interpretation of the shape of an object at each level of abstraction. In particular, we shall examine surface structure in terms of differential features and see how this structure can be used to facilitate solution of the motion and

geometric inference problems.

2.2 Computing 3D Descriptions: Formal Framework

In this section we present an approach to shape in a domain of piecewise smooth, solid objects. The shape of an object is defined in this context as a hierarchical set of descriptions (figure 3).

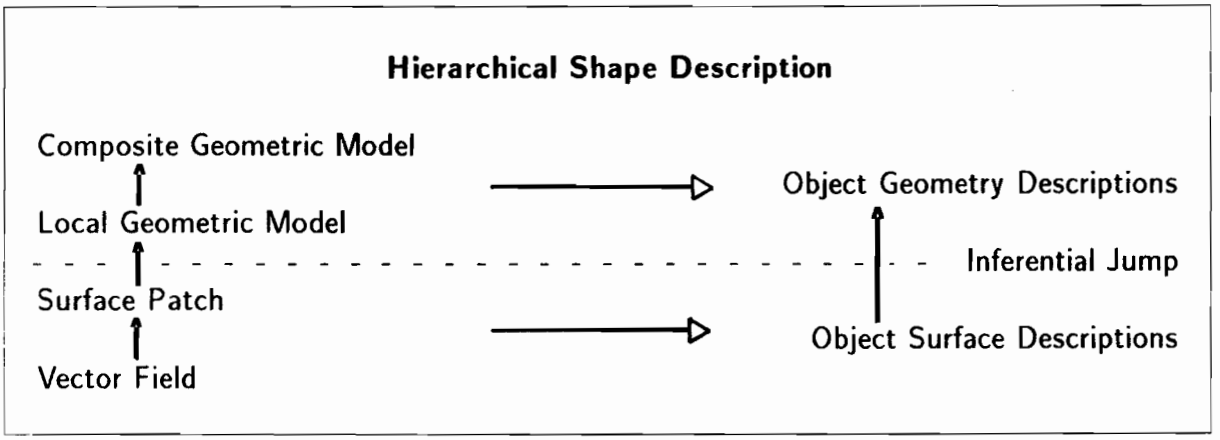

Figure 3: The Shape Hierarchy

The shape hierarchy consists of two basic levels:

1. A description of all visible surfaces of the object and their intrinsic properties as defined by differential geometry.
2. A description of the object as a conjunction of volumetric primitives (in this case ellipsoids and cylinders) that are inferred from the visible surface descriptions.

Object surface descriptions are represented by a structure which we call a surface graph. The nodes of the graph are patches of surface that are assumed to correspond to volumetric primitives that comprise an object (figure 4). A patch consists of a set of intrinsic feature vectors that describe pertinent characteristics of the surface at a local level. This local structure provides the basis from which the global geometry of the patch is inferred by applying

constraints from differential geometry, hence the indication of an *inferential jump* in figure 3. In other words, these constraints are used to *select* an appropriate choice of primitive component that best reflects the characteristics of a particular surface patch. The local geometric model is subsequently parameterized from this same data. Once the set of primitives that define an object has been computed, a composite description is obtained by computing its conjunction.

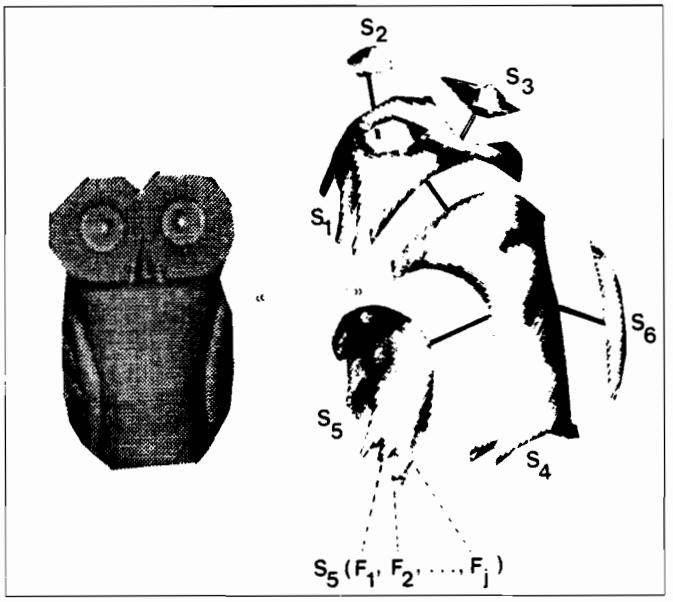

Figure 4 The surface graph representation of an object

In the following sections we will consider in detail what a surface graph represents and how it is computed. Before doing so, however, a number of assumptions will be made concerning the process of computing the shape model outlined above. First it is assumed that all surfaces of an object are presented to the viewer in a sequence of views of unknown relationship to each other, and that the loci of points describing them are computable (ultimately as a depth map). We also assume that the problem of associating surfaces to particular objects has been resolved (i.e. we can determine which regions in an image correspond to which objects). Finally, we have to make some assumptions about the geometry of objects in our domain, as it is clear that conjunctions of ellipsoids and cylinders have limited scope. We thus restrict our domain to objects formed by deformations of a sphere in which no surfaces are allowed

to intersect. These deformations are further limited to second order surfaces; however, we can maintain some flexibility by placing no restrictions on how neighbouring deformations are joined. We will begin by considering a representation of an object's surfaces contained in a single viewpoint, and later consider how to augment it with information from other positions.

2.2.1 Representing Surfaces in a Single View

The fundamental representational problem is *what aspects of surfaces should be made explicit*? In the case of polyhedral objects, the answer is in terms of edges and vertices. These are *differential* properties from which the entire scene could be described. A similar line of reasoning would therefore lead one to conclude that a differential approach to describing curved surfaces should be appropriate, implying a view of structure in terms of differential geometry. In our present case, the differential structure of a surface is used to:

1. Identify loci on the surface where the intrinsic nature of a surface undergoes change. This serves as a form of "parts decomposition" [5] for curved surfaces which is used to decompose a surface into patches.
2. Infer the more global properties of a surface patch, in particular its geometric interpretation. This forms the basis for the geometric inference problem.
3. Serve as tokens in correspondence computation. In order to augment the surface representation with information from other views, correspondence must be computed between adjacent view surfaces. Differential properties which are intrinsically related to a surface are ideal tokens as they are independent of viewpoint.

As we shall see in the following sections, differential geometry provides the necessary mathematical tools for extracting the structure needed for the above tasks.

---

[5] Which we refer to as "curved edges".

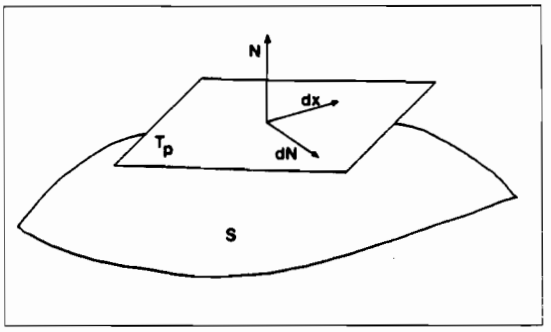

Figure 5 A surface and its tangent plane

2.2.1.1 A Differential View of Surfaces

Let a surface S[6] be defined as a continuous function of order ≥ 2 of the form $z = f(x, y)$, where z represents the height of the surface at a point (x, y) from a reference plane z_r parallel to the view plane of the viewer. Let T_p be defined as the plane tangent to a point p lying on S at position (x_p, y_p), and N_p as the unit normal vector to the surface at that point (figure 5). One of the most important concepts of differential geometry is that of *normal curvature*, denoted by the symbol Π and defined as $\Pi \stackrel{\triangle}{=} \langle -d\bar{x} \cdot d\bar{N} \rangle$, the value of a normal section oriented in direction $d\bar{x} = (du, dv)$ in the tangent plane T_p [do Carmo 1976] (figure 5). Π , also known as the *second fundamental form* of differential geometry, can be conveniently expressed in terms of a surface S:

Let

$$e = \frac{f_{xx}}{\sqrt{1 + f_x^2 + f_y^2}} \qquad f = \frac{f_{xy}}{\sqrt{1 + f_x^2 + f_y^2}} \qquad g = \frac{f_{yy}}{\sqrt{1 + f_x^2 + f_y^2}},$$

then

$$\begin{aligned} \Pi &= e\,du^2 + 2f\,du\,dv + g\,dv^2 \\ &= (\,du \quad dv\,) \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} du \\ dv \end{pmatrix} \\ &= \xi' A \xi. \end{aligned} \tag{1}$$

---

[6] This should actually be written as S(t). However, in order to simplify notation we shall assume that the time dependency is implicit in S.

The behavior of Π in the neighbourhood of a point p provides certain insight into making deductions about the nature of S as it moves away from the intersection with the tangent plane T_p. In particular, Π takes on minimum and maximum values which can be shown to be the eigenvalues of A in the matrix representation of Π . The directions associated with the minimum and maximum curvatures can also be shown to align with the eigenvectors of A [do Carmo 1976], and are called the principal directions. The connection of Π to surface characteristics is made through Gaussian curvature, which is the product of the minimum and maximum curvatures. The sign of Gaussian curvature indicates whether the surface is elliptical (+), hyperbolic (-), or parabolic (zero Gaussian curvature) in the neighbourhood of a point p. Of particular interest are those points on S where curvature changes either continuously or discontinuously. Such changes in Gaussian and Π curvature provide a notion of "edges" for curved surfaces. We use the term *curved edge* to refer to a locus of points marking either a continuous or discontinuous change on a surface.

The surface characteristics we are interested in are marked by discontinuities in the surface function and changes in sign or local maxima (peaks) in curvature at so-called critical points on a surface. We define a "patch" to be a set of points on a surface delimited by a curved edge comprised of these critical points. Thus, a surface patch forms a homogeneous region on a surface from which we attempt to infer more global properties of the surface, for example, the volumetric primitive on whose surface the patch is assumed to lie. Of particular interest to the present research is the significance of curved edges formed by local maxima in Π curvature. As Richards and Hoffman [1984] point out, the intersection of two convex bodies will be marked by local maxima in Π curvature. The implication for a Generalized Cylinder representation of an object is that local maxima in Π serve to mark the locations of limbs that decompose the object into component cylinders (for example, see figure 6).

2.2.1.2 The Surface Graph

We describe a surface S by a representation which we call a *surface graph* G. As is

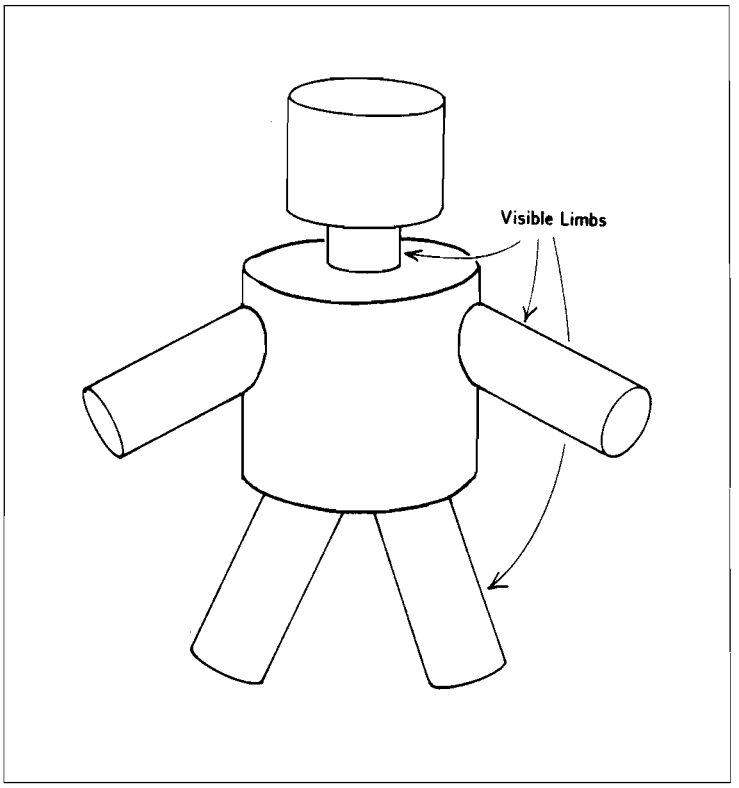

Figure 6 Limbs marked by extrema in curvature

shown in figure 4, a surface is described in terms of a graph structure composed of a set of non-overlapping patches S_i. The object O implied by $\bigcup_i S_i$ is assumed to be the conjunction of a set of convex volumetric primitives V_i, i.e.

$$O = \bigcup_i S_i \approx V = \bigcup_i V_i \qquad (2),$$

where each S_i is assumed to lie on the surface of V_i[7]. The surface regions corresponding to each S_i can be computed by decomposing surface S along the curved edges marked by discontinuities in the surface function or inflections and local maxima in Π . Although the identification of extrema in Π on S can be accomplished by straightforward analysis, the location of contours that form patch boundaries is a difficult problem in practice. In fact,

[7] This condition is actually relaxed to include cases where V_i is composed of several surface patches. A cylinder, for example, would be composed of three patches, two of which are parabolic. The non-parabolic patch, however, still contains adequate structure such that a cylindrical volume may be correctly inferred and parameterized.

it is precisely the same low-level grouping problem that occurs when tracing edges given the discontinuities in an intensity array [Leclerc & Zucker 1984; Leclerc 1986]. Such problems have been addressed in a computational theory developed by Zucker [1982] and its implementation by Zucker and Parent [1982,84].

G is defined in terms of the decomposition of S, where each node corresponds to an S_i and the links between nodes represent the adjacency relationships among the S_i. Each patch forms a compact representation for a surface neighbourhood, since each in turn may be represented by an appropriate parameterization. However, patch descriptions are still global in nature and must be supplemented by more local descriptors. This is especially important when correspondence of different views is to be considered. For this reason, we describe each S_i further as $\bigcup_j F_j$, where F_j is a vector of intrinsic surface features, listed below, that is associated with each point $p_j \in S_i$. By applying standard analysis techniques (described in Chapter 3) to a set of points representing a surface in 3D, the following features may be computed on a pointwise basis:

- Principal curvatures.
- Directions associated with principal curvatures.
- Discontinuities in the surface function.
- Inflection points in the curvature of a surface.
- Peaks in the curvature of a surface.
- Whether a surface is convex or concave at a point.
- A labeling of the surface type: elliptical, hyperbolic, or parabolic.

In addition to components already described, each S_i also includes a set of boundary relations which point to the neighbouring patch surfaces. We will next consider how G is augmented to include descriptions of surfaces from adjacent views, and then discuss how a composite geometric description can be obtained from the complete graph.

2.2.2 Aggregation from Different Views

In discussing the surface graph G above, we made the explicit assumption of the existence of a surface S of the form $z = f(x, y)$. This form characterizes a viewer-centered description of the visible surfaces of an object such as might be obtained from a depth map. However, a full description of an object requires that *all* surfaces be presented to the viewer through a sequence of views, and that these be expressed in a common coordinate system. What is required, then, is a linear[8] transformation T that maps $f(x, y)$ into $W(x, y, z)$ where W defines a suitable world coordinate system. This problem is easily solved if for every pair of adjacent views, an inter-frame transformation T_i is computable that maps $f_{i-1}(x, y)$ into $f_i(x, y)$. Then, given that the system is linear, the transformation mapping any particular frame a into some other frame b is given by the composite mapping,

$$T_{ab} = \prod_{a+1}^{b} T_i. \tag{3}$$

The principal difficulty in the above approach is in the computation of the inter-frame transformations T_i. As was discussed briefly in the background section, most approaches to this problem attempt solution by exploiting nonlinear constraints relating projected surface configurations to object motion parameters. The situation in the present case, however, is not as complex because point correspondences are between *surfaces* in 3D rather than between *projections* in 2D. This yields the following linear relation:

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{pmatrix} \begin{pmatrix} X_{i-1} \\ Y_{i-1} \\ Z_{i-1} \\ 1 \end{pmatrix}, \tag{4}$$

where $\langle X_i, Y_i, Z_i \rangle$ and $\langle X_{i-1}, Y_{i-1}, Z_{i-1} \rangle$ are corresponding points on S in frames i and $i-1$ respectively.

---

[8] We have assumed that objects remain rigid during the inter-frame interval.

In the ideal case where correspondences are known exactly, solution of the above equation reduces to solving four sets of four simultaneous equations for four pairs of non-coplanar point correspondences. In practice, however, such simplistic solutions do not suffice because the location of surfaces and point correspondences are both subject to error. If a reasonable number of estimates are available, however, and the error is assumed to be normally distributed, then a linear regression approach to the above estimation is appropriate [Johnson & Wichern 1982]. Notice in (4) that X_i, Y_i, and Z_i are linear combinations of elements of the rows of T_i, which are by definition linearly independent. Thus the classical regression model shown below can be applied separately for each row of T_i and the results concatenated.

$$\mathbf{A} = \mathbf{B}\beta + \epsilon \tag{5}$$

For example, application of the above model to estimating the first row of T_i yields

$$\begin{pmatrix} X_{i,1} \\ \cdot \\ \cdot \\ \cdot \\ X_{i,n} \end{pmatrix} = \begin{pmatrix} X_{i-1,1} & Y_{i-1,1} & Z_{i-1,1} & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ X_{i-1,n} & Y_{i-1,n} & Z_{i-1,n} & 1 \end{pmatrix} \begin{pmatrix} t_{11} \\ t_{12} \\ t_{13} \\ t_{14} \end{pmatrix} + \begin{pmatrix} \epsilon_1 \cdot \\ \cdot \\ \cdot \\ \epsilon_n \end{pmatrix}. \tag{6}$$

One method of obtaining an estimate $\hat{\beta}$ for β is by applying least squares methods to minimizing ϵ. Thus,

$$\hat{\beta} = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{A}. \tag{7}$$

The accuracy of a regression approach to parameter estimation will be highly dependent on the point correspondences that are at its basis. The major difficulty in computing correspondence is the problem of localizing unique tokens across adjacent views. For this reason, much emphasis has been placed on the location of tokens such as corners [Kitchen & Rosenfeld 1980], zeros in mean curvature [Dreschler & Nagel 1982], and other features of image intensity [Nagel 1983]. With curved surfaces, however, such cues are not as apparent, so a different notion of structure is required. This is precisely what the surface graph G makes explicit, as each point on the surface is represented by the set of intrinsic feature vectors

F_j. Furthermore, to better cope with noise and quantization error, correspondence can be extended to include the structure of a local neighbourhood in terms of the set of F_j defining it.

Definition: To each coordinate $p_j = (x_j, y_j)$ of an $m \times n$ neighbourhood on a surface S, associate a feature vector F_j. This set then defines a feature field $\mathbf{F} = \bigcup_j p_j$ in an $m \times n$ neighbourhood of S centered at coordinates (x_k, y_k) with orientation θ.

Let $F_{t-1,j} = \langle \alpha_{j,1}, \alpha_{j,2}, ..., \alpha_{j,n} \rangle$ be an element of an $m \times n$ feature field $\mathbf{F}_{t-1}$ centered at coordinates (x_k, y_k) with orientation θ in frame $t-1$. Let $F_{t,j} = \langle \beta_{j,1}, \beta_{j,2}, ..., \beta_{j,n} \rangle$ be an element of an $m \times n$ feature field $\mathbf{F}_t$ centered at coordinates (x'_k, y'_k) with orientation θ' in frame t. Correspondence is performed by maximizing the following functional:

$$\max_{x',y',\theta'} \left\{ \sum_n w_n \frac{\sum_j \alpha_{j,n} \beta_{j,n}}{\sqrt{\sum_j \alpha_{j,n} \alpha_{j,n}} \cdot \sqrt{\sum_j \beta_{j,n} \beta_{j,n}}} \right\} \tag{8}$$

where w_n is a component of the weighting vector W allowing for variable emphasis of features.

The functional, (8), is a normalized cross correlation of the two feature fields $\mathbf{F}_{t-1}$ and $\mathbf{F}_t$, which returns a maximum value of 1 for a perfect match. Correlation is performed in a three-parameter space (x', y', θ') that accounts for the rotation and translation of a local neighbourhood as viewed by a stationary observer. Besides indicating a best match between two neighbourhoods, the functional also serves as a measure of goodness for a particular match. In this manner, poor matches, such as those that might arise from self-occlusion, or severe foreshortening of a surface, are eliminated from consideration in computing an estimate to the inter-frame transform, T_i. The result of computing this correlation over M different neighbourhoods will be a set of N (where $N \leq M$) point correspondences from which the inter-frame transform T_i can be estimated.

In addition to maximizing the correlation between feature fields, one further check can be made on the appropriateness of a match. This consists of verifying the consistency of

the feature labels associated with the corresponding neighbourhood origins. A match can be disallowed, for example, in the case where surface labelings (e.g. ellipsoidal, hyperbolic, parabolic) do not match between a candidate and its match.

Another problem that must be dealt with is the selection of candidate neighbourhoods for matching. It is obvious that matching will have a higher probability of success in neighbourhoods that are highly differentiated; thus, regions with extremal features are desirable candidates. These are already made explicit in G in terms of peaks and inflections in Π .

Once the computation of correspondence and the subsequent estimation of the set of T_i are accomplished for a sequence of images, a composite surface graph G^c is constructed. This involves mapping each G_i into some world reference frame and merging its component S_i's to the existing structure as expressed below:

$$G^c = G^c_{i-1} \bigcup_i G_i - G^c_{i-1} \bigcap_i G_i. \tag{9}$$

The fundamental constraint employed above is that only one surface patch may occupy any location in 3D. Overlapping patches are deemed to be multiple instantiations of the same surface. In addition to adding new S_i's to G^c, the merging process must also take into account occlusion by allowing the modification of existing, but occluded, patches as hidden surfaces become visible when new views are added. The particular choice of reference frame for the reconstruction implied by (9) is one of convenience since the orientation of the object in 3D is not known. Without loss of generality, we choose to map all views into the coordinate system of the first or last frames by applying the successive transformations shown in (3).

2.2.3 Geometric Inference: From Surfaces to Objects

The next stage of representation involves the abstraction of surfaces into a composite geometric model comprised of volumetric primitives inferred from G^c. We call this the process of *geometric inference*. The basic assumption is that all objects of interest within a domain

can be adequately represented by a conjunction of suitable primitives. In the present domain, objects are assumed to be composed of ellipsoids and cylinders. The paradigm consists of the following steps:

1. Traverse G^c and deduce from the contour, shape and curvature characteristics of each S_i the most likely primitive V_i whose surface the patch is assumed to lie on[9].
2. Compute the parametric description for each inferred primitive.
3. Compute the conjunction of all inferred primitives. A volumetric representation of the object is obtained by computing:

$$V^o = \bigcup_i V_i - \bigcap_i V_i. \tag{10}$$

A prerequisite for fitting models such as ellipsoids and cylinders to data is the determination of the axes along which parameterization takes place. In [Agin & Binford 1976], for example, the authors relied on symmetry cues in locating the axes of cylinders which they fit to range data. We refer to this as the problem of determining the intrinsic coordinate system (or *ICS*) of a set of points defining some object in 3D. In regular geometric structures such as ellipsoids and cylinders, the *ICS* is usually defined along axes of symmetry[10]. From the point of view of analytical mechanics, determining the *ICS* is equivalent to finding the axes of rotation that minimize the angular momentum of a body rotating in 3D (the dynamic balancing

---

[9] The scope of the deduction process is expanded to include the characteristics of additional patches should V_i be composed of more than one.

[10] This is one reason for decomposing an object into a set of volumetric primitives, because it enables us to compute the orientation of an object locally. The global orientation is subsequently determined by aggregating local structures along the lines suggested by Marr [1982] for generalized cylinder representations of objects.

problem) [Fowles 1970]:

$$\mathbf{L} = \mathbf{I}\omega \qquad \text{where} \qquad \begin{array}{l} \mathbf{L} = \text{angular momentum} \\ \mathbf{I} = \text{the intertia tensor} \\ \omega = \text{rotations about axes} \end{array} \tag{11}$$

Observe that if **I** were diagonal, then the coordinate (x, y, z) axes would correspond to the intrinsic coordinate system. Rewriting the above equation as:

$$\begin{bmatrix} \lambda\omega_1 \cos\alpha \\ \lambda\omega_2 \cos\beta \\ \lambda\omega_3 \cos\gamma \end{bmatrix} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_1 \cos\alpha \\ \omega_2 \cos\beta \\ \omega_3 \cos\gamma \end{bmatrix} \tag{12}$$

The eigenvectors of **I** are the direction cosines of the ICS

$$\bar{x}' = \begin{pmatrix} \cos\alpha_1 \\ \cos\beta_1 \\ \cos\gamma_1 \end{pmatrix} \quad \bar{y}' = \begin{pmatrix} \cos\alpha_2 \\ \cos\beta_2 \\ \cos\gamma_2 \end{pmatrix} \quad \bar{z}' = \begin{pmatrix} \cos\alpha_3 \\ \cos\beta_3 \\ \cos\gamma_3 \end{pmatrix}, \tag{13}$$

where $\bar{x}'$, $\bar{y}'$, and $\bar{z}'$ are the axes of the *ICS*.

In order to perform the above computations, the inertia tensor **I** must be estimated from the surfaces defining a particular component. In the case where the set of surfaces is incomplete, such as where only a limited number of views are available, the resulting *ICS* will be meaningless unless enough of the original symmetry is preserved. For example, consider the case of a single view of a cylinder parallel to the viewing plane. The direction of major axis will be estimated correctly, however, it will be displaced in depth. In this particular example, having the direction of the major axis and a sample of points on the surface is sufficient to recover the parameters of the cylinder. But this will not be the situation in general, and is why we require information from multiple views to recover the complete surfaces of an object.

Once the axes are determined and models selected for each patch, parameterization can be accomplished with standard approximation methods such as least squares. Finally, the geometric inference paradigm is completed by computing the intersection of all parameterized components. Any further interpretation will depend on the context of a particular domain.

2.2.4 Summary

In this chapter we have outlined a model for characterizing the three-dimensional shape of a moving object. Shape is represented as a hierarchical structure with two major levels. The first is a representation for the surfaces of an object, as presented to the viewer in a sequence of views, and the second a representation of the object as a conjunction of volumetric primitives. Our approach calls for exploiting the local properties of a surface, such as functions of surface normal curvature, to provide descriptions of a surface. These are subsequently used to decompose the surface into homogeneous regions from which inferences about local geometric structure can be made (i.e. which volumetric primitive best describes a particular surface region).

Because a single view of an object is often inadequate for purposes of reconstruction, some provision must be included for aggregating the surface model with information from other viewpoints. The same local descriptors that are used for surface decomposition can also be exploited in solving the correspondence problem. This enables the estimation of the object's motion in between different views, and is a pre-requisite for reconstructing the complete surfaces of the object. The complete surface model, which we call the composite surface graph G^c, can be abstracted as a set of volumetric primitives by making use of constraints from differential geometry. The conjunction of these primitives serves as a composite geometric representation for the object that can be used for purposes of interpretation and feature measurement.

Our shape model is a synthesis of many ideas that have been around Computer Vision for the past decade. The major contribution of this work, however, is in identifying the global constraints that tie the component sub-problems together, and using these constraints to derive a computational model that forms the basis of a working computer vision system. This chapter sketched out the components of this model in an attempt to give the reader a flavour of the concepts involved. The following three chapters are each devoted to the individual

components that define our shape problem and provide the necessary detail that is lacking in an overview.

Chapter 3 Computing Surface Descriptions from a Single View

It is quite remarkable that humans can almost instantaneously recover the shape of objects in a scene from a casual glance. This chapter will focus on the problem of how the surfaces of an object are computed from an image, and how a description of these surfaces can be computed in terms of the surface graph G.

3.1 Estimating Local Surface Orientation: Shape from Shading

The general aim of "Shape-from-X" approaches in vision is the inversion of the process of image formation which is inherently underdetermined. In the following section we will investigate one of these approaches, shape from shading. We are motivated by two considerations: shading cues play an important role in the interpretation of smooth surfaces [Pentland 1982a], and our work with images from light microscopy. The obvious question, given our stated interest in the differential properties of surfaces, is what can be derived from shading about the local structure of surfaces.

Smith [1983a] shows that local analysis of shading can at best provide only relative surface curvature, and that in general, surfaces cannot be recovered in this manner. On the other hand, Pentland [1982a,b] has shown that by constraining surfaces to being locally spherical, qualitative estimates of shape can be recovered. In our work with images from light

microscopy, we observed that shading cues predominate so that humans are able to make reasonable estimates of shape (at least compared to the distortion induced by the imaging process when object sizes tend towards the wavelength of the light source). This led us to consider Pentland's approach and to analyze when the distortion induced by the assumption of locally spherical surfaces would be tolerable in reconstruction. We begin by considering the process of image formation using a simplified model [Horn 75,77; Pentland 82a,b], and then the conditions under which reasonable estimates of local surface orientation can be derived with this model.

3.1.1 A Simplified Model of Image Formation

The process by which photographic images are created provides a starting point in deriving a model for image formation. Direct illumination from a light source, and secondary illumination from nearby objects, are reflected off the surface of an object and are directed onto a photo-sensitive film by an optical lens (figure 7). The relative positions of object, light sources, and camera position affect the intensity distribution recorded on film and thus the composition of the image. By considering only a single light source and ignoring secondary illumination, a simplified model can be obtained (figure 8) [Horn 1970,75,77]. Let N be a unit normal vector to a planar patch of surface, V the direction of view of a camera or observer, and L the direction of a light source infinitely distant from the surface. The relationship of these components to the reflected intensity I of the patch is given by the following equation [Pentland 1982a] for a surface in orthographic projection to the viewer:

$$I = \rho\lambda(N \cdot L)R(N,L,V)(N \cdot V)^{-1} \tag{14}$$

where ρ = surface albedo

λ = incident light flux

$(N \cdot L)$ = illumination foreshortening term

$R(N,L,V)$ = surface reflectance function

$(N \cdot V)^{-1}$ = projective scaling term

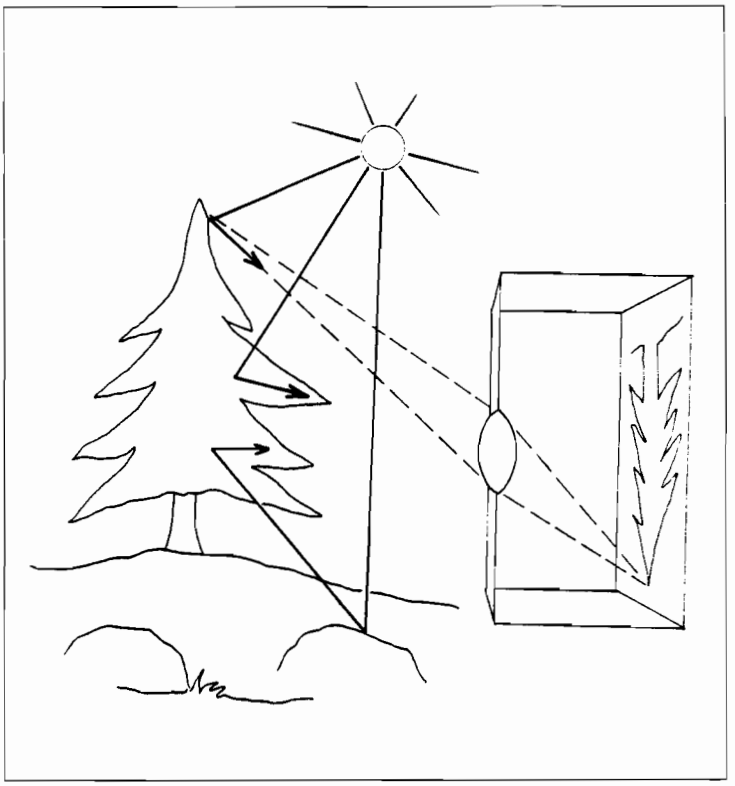

Figure 7 Camera model of image formation

λ represents the amount of light flux per unit area that impinges on the surface patch. This flux is then scaled according to the angle between the light source and surface normal $N \cdot L$. The light that is eventually reflected back to the viewer is further scaled by three factors: (1) The surface albedo ρ, an intrinsic property of a surface that defines the amount of incident light flux that is reflected from the surface; (2) $R(N, L, V)$, the surface reflectance function [Horn 1970,75,77]; and (3) $N \cdot V$ a second foreshortening term that accounts for projective scaling of reflected intensity into the viewing plane.

The reflectance function $R(N, L, V)$ expresses the reflectance of a surface as a function of direction and varies widely between two extremes. At one end a surface is Lambertian, in which light is reflected evenly in all directions, typical of most mat surfaces. At the other, a surface is mirror or specular in which case light is reflected in but a single direction, equal to the incident angle. Horn [1977] has suggested a general model for $R(N, L, V)$ which combines terms representing both extremes. However, if we avoid the consideration of specular surfaces,

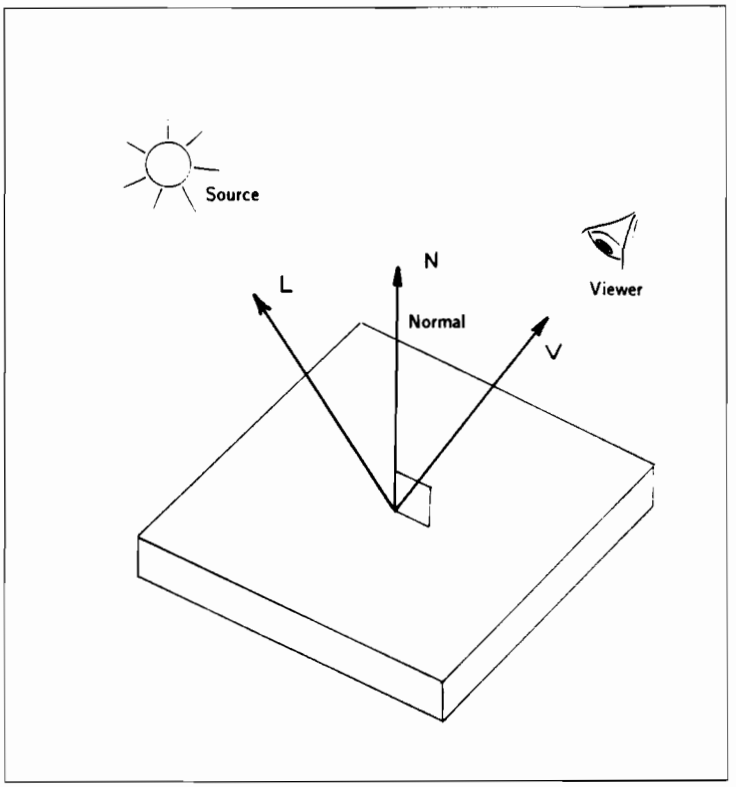

Figure 8 Simplified model of image formation

things simplify considerably. This was the approach taken by Pentland [1982a,b] in performing his analysis of shape from shading. The reflectance function for a Lambertian surface is

$$R_L = (N \cdot V) \tag{15}$$

Substituting this term into the image irradiance equation (14) we obtain

$$I = \rho\lambda N \cdot L \tag{16}$$

The simplified irradiance equation (16) is a function of six unknowns, ρ, λ, and two parameters each defining the unit vectors N and L. Pentland's approach to solving (16) was to employ constraints from differential geometry in the local neighbourhood of N by looking at derivatives of I in reducing the number of unknowns. We took a different approach, similar to that of [Lee & Rosenfeld 1983] by assuming that the position of the light source L was either known or else could be estimated.

For convenience, we assume that the light source direction L is parallel to the Z axis. Since we know the *actual* direction L_a, we can always apply a transformation mapping N into N_a upon computing N in what Lee and Rosenfeld [1983] refer to as *light source coordinates*. If L is assumed to be parallel to the Z axis, then $L =< 0,0,1 >$ and (16) simplifies to:

$$I = \rho\lambda N_z \tag{17}$$

where N_z is the Z component of the surface normal vector N.

A common method for expressing N is through the use of two angles, tilt τ, and slant σ that define N in relation to the plane of the viewer [Stevens 1979, Witkin [1980], Pentland 1982a,b]. They are defined as follows:

$$\sigma = \cos^{-1}(N_z) \qquad \tau = \tan^{-1}\left(\frac{N_y}{N_x}\right) \tag{18}$$

Notice that the slant angle σ is completely determined from (17) provided that the proportionality constant $\rho\lambda$ can either be measured or determined.

It remains to estimate the tilt angle τ, which together with σ, completely determine the unit normal vector N. We begin by examining the relationship between dN_z and the surface tilt τ for a surface $z = f(x,y)$ in orthographic projection to the view plane. The unit normal N to z at a point p, $z = f(x_p, y_p)$, is given by:

$$N = \frac{\langle f_x, f_y, -1\rangle}{\sqrt{1 + f_x^2 + f_y^2}} \tag{19}$$

From (19) it follows that

$$\tau = \tan\left(\frac{f_y}{f_x}\right)^{-1} \quad and \quad N_z = \frac{-1}{\sqrt{1 + f_x^2 + f_y^2}} \tag{20}$$

Applying the chain rule to the expression for N_z results in the following:

$$dN_z = \frac{\partial N_z}{\partial x}\partial x + \frac{\partial N_z}{\partial y}\partial y$$

$$dN_z = -\left\{ \frac{f_x f_{xx} + f_y f_{yx}}{(f_x^2 + f_y^2 + 1)^{\frac{3}{2}}} \partial x + \frac{f_x f_{xy} + f_y f_{yy}}{(f_x^2 + f_y^2 + 1)^{\frac{3}{2}}} \partial y \right\} \tag{21}$$

It is instructive to compare (21) to dz, the directional derivative on the surface $z = f(x, y)$:

$$dz = f_x \partial x + f_y \partial y \tag{22}$$

Comparison of the two equations shows that dz will be aligned with dN_z *if and only if*:

$$\frac{f_y}{f_x} = \frac{f_x f_{xy} + f_y f_{yy}}{f_x f_{xx} + f_y f_{yx}} \tag{23}$$

Notice, however, that dN_z is related to the first directional derivative of intensity by the following:

$$dI = \rho \lambda dN_z \tag{24}$$

Thus, (23) provides the necessary and sufficient conditions under which dI is aligned with dz. Under these conditions it is also clear that the surface gradient direction, which is by definition τ, is aligned with the intensity gradient. One can show, for example, that (23) is satisfied for a spherical surface of the form $f(x, y) = \sqrt{R^2 - x^2 - y^2}$ since

$$\frac{f_y}{f_x} = \frac{f_x f_{xy} + f_y f_{yy}}{f_x f_{xx} + f_y f_{yx}} = \frac{y}{x} \tag{25}$$

It would appear from the foregoing analysis that *exact* recovery of surface orientation from image intensity is unlikely, in agreement with Smith [1983a]. Even if the light source direction vector L and albedo term $\rho\lambda$ were known or could be estimated locally, the constraints imposed by (23) effectively limit consideration to a limited class of surfaces. Does this mean that shape from shading approaches are untenable in surface recovery? The answer to this question depends on how much distortion one is willing to accept in estimates of local surface orientation. Pentland [1982a] showed that humans make errors in orientation estimates based *solely* on shading cues, for example in underestimating surface relief. These considerations led us to look at the problem from a different perspective: *given that local surface estimates will be in error, how does the resulting distortion affect the computation of shape, and under*

what circumstances is this error acceptable? In the next section we examine this question and show that quite reasonable estimates of shape can be recovered from local analysis of shading.

3.1.2 Error Components in Local Shading Analysis

In order to reconstruct the shape of an object, it is important to understand how each of the component processes affects the *overall* recovery of shape. Errors in estimating local surface orientation arise from three sources:

- Uncertainty in parameter estimation: the light source direction vector L and the albedo term $\rho\lambda$.
- The shape of the local neighbourhood does not fulfill the conditions required by (23).
- Error in determining the direction of the intensity gradient of the local neighbourhood.

How then does one investigate each of the above component effects on the global recovery of shape? Our approach was to look at the effects of each of these components on the recovery of scene generation parameters [11]. We applied the following procedure:

1. Generate front and rear view images of a geometric shape from a parametric description.
2. Apply local shading analysis to each of the images to recover surface orientation at each point.
3. Integrate the resulting vector fields to yield depth maps corresponding to each view.
4. Reconstruct the complete surface description by mapping each view into a common coordinate system (i.e. piece the front and rear views together).
5. Compute a parametric description of the object from the resulting reconstruction.

---

[11] Images of objects were generated from parametric descriptions of geometric shapes.

The above procedure, which is in fact a subset of our paradigm for describing shape, provides a means of experimenting with individual processes in recovering global descriptions. In this manner we can judge the significance of a particular component, and the conditions under which error in that component might be tolerable in computing a global description. Notice that we have accepted the fact that *exact* recovery of surfaces is unlikely. What we want to know is how good an estimate can be obtained given the limitations of local shading analysis.

3.1.2.1 Estimating L and $\rho\lambda$

The first step in estimating local surface orientation is the determination of the illuminant direction L for the surface, and constants $\rho\lambda$, for a particular estimation neighbourhood. Pentland [1982a], in the spirit of Witkin [1981], suggested the assumption of an isotropic distribution of surface normals, an assumption valid for most surfaces. By (24) it can be seen that the gradient directions on a surface, dI, are a linear combinations of the illuminant direction L. Since the normals are assumed to be isotropically distributed, so too are the gradient directions on the surface. Thus, Pentland suggested using a maximum-likelihood procedure in estimating the light source direction vector L, and obtained the following result:

$$(\hat{L}_x \quad \hat{L}_y) = (\beta' \quad \beta)^{-1} \beta' (d\bar{I}_1 \quad d\bar{I}_2 \quad \ldots \quad d\bar{I}_n) \tag{26}$$

where $\hat{L}_x$ and $\hat{L}_y$ are the non-normalized x and y components of the light source direction vector L; β, a $2 \times n$ matrix of directions (dx_i, dy_i); and $d\bar{I}_i$, the means of dI over the estimation region along each of n image directions (dx_i, dy_i). The full illuminant direction is given by

$$L_x = \frac{\hat{L}_x}{k} \qquad L_y = \frac{\hat{L}_y}{k} \qquad L_z = \sqrt{1 - L_x^2 - L_y^2}$$

where

$$k = \sqrt{E(dI^2) - E(dI)^2} \tag{27}$$

The $\rho\lambda$ term in (24) is accounted for by parameter k above. Pentland addressed the problem of estimating $\rho\lambda$ by introducing a further assumption, that the estimation region for L was

spherical. From this he was able to show that an estimator for $\rho\lambda$ was the standard deviation of dI along any particular image direction (dx_i, dy_i) [Pentland 1982a]. An alternate approach to estimating these parameters is presented in [Lee & Rosenfeld 1983], based on Pentland's assumptions.

We applied Pentland's estimator for the light source direction L to the image of a sphere (figure 9a). This eliminates any concern over the surface not satisfying (23) or not being able to determine the gradient direction accurately. The $\rho\lambda$ term was estimated by simply taking the min/max of the intensity distribution over the surface and scaling slant such that minimum and maximum intensities corresponded to slant angles of 90° and 0° respectively. The image was artificially generated with a mean error in the intensity distribution of 3.12%, a standard deviation of 5.78%, and a maximum error of 19.55%. Figure 9b shows the resulting distribution of surface normals, and figure 9c the reconstructed surface for the sphere image. To recover the generating parameters for the sphere, we fit an ellipsoid to the reconstructed surface and observed the ratios of the major axes. One would expect identical axis lengths for a perfect sphere, and deviations from unity to indicate a measure of "roundness". Results are summarized in Table 1 (Appendix C).

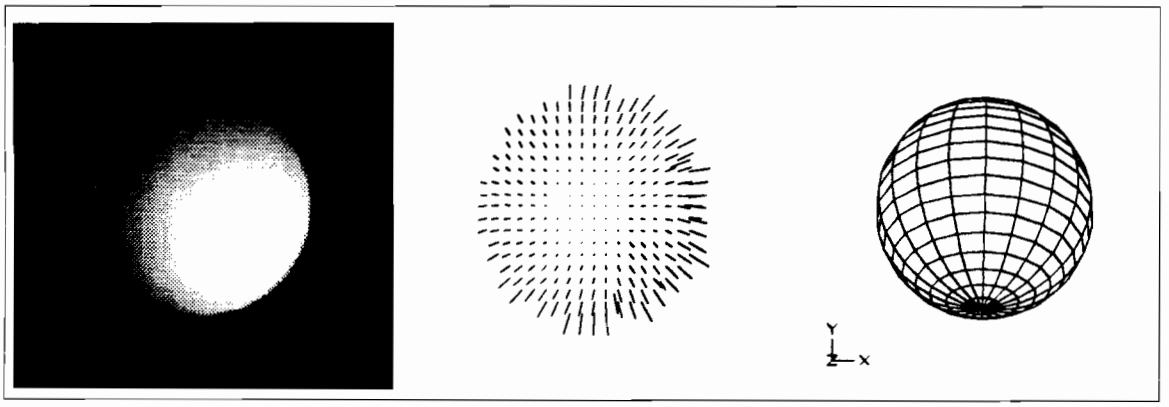

Figure 9a-9c Sphere reconstruction: (a) original image (b) estimated surface normals (c) reconstructed surface

The above experiment was repeated for different light source positions and confirmed the findings of Lee & Rosenfeld [1983] with regard to the accuracy of Pentland's L position

estimator. The emphasis of our experiments, however, was on obtaining upper bounds on reconstruction error for a given uncertainty in light source position L. We assumed that L was known to within 15° to 20° solid angle, and found that quite precise recovery of scene generation parameters was indeed possible. In fact, as can be verified by Table 1, L position errors of up to 20% have little effect on parameter recovery for a sphere. Three light source positions were used in the reconstruction process: the exact position used to generate the image, the estimated position computed with Pentland's procedure, and a "forced" position selected such that L was at maximum tolerance. Notice that the ratios of axis lengths, which must be unity for a sphere, are within 5% for all L positions.

3.1.2.2 Effects of Non-Spherical Surfaces

The second major error component that must be considered in estimating local surface orientation is the effect of assuming the alignment of the intensity gradient, dI_{max}, and the surface tilt angle, τ, when the surface does *not* satisfy (23). A good strategy to pursue in studying this effect is to consider how tilt estimates are distorted as a spherical surface is deformed into either an elliptical or hyperbolic surface. We begin by considering functions of the form

$$f(x,y) = c\sqrt{\pm(1 - \frac{y^2}{b^2} - \frac{x^2}{a^2})} \tag{28}$$

where the discriminant is positive for an elliptical surface and negative for a hyperbolic surface, both of which are centered at the origin with principal axes, a, b, and c, parallel to the coordinate axes.

If we apply equation (23) to the above, we find that the estimated tilt angle, $\hat{\tau}$, is given by

$$\hat{\tau} = -\tan^{-1}\left\{\frac{\left(\left(b^2 - a^2\right)x^2 + a^4\right)y}{\left(b^2 - a^2\right)xy^2 - b^4x}\right\} \tag{29}$$

for *both* elliptical and hyperbolic surfaces. The actual tilt angle τ, again for both surfaces, is

$$\tau = \tan^{-1}\frac{a^2y}{b^2x} \tag{30}$$

Figure 10 shows a plot of tilt estimation error, i.e. $|\hat{\tau} - \tau|$, as a function of eccentricity ratio $\frac{a}{b}$, obtained from equations (29) and (30). A ratio of 1 corresponds to a purely spherical surface, with large ratios tending towards cylindrical surfaces. Notice that for even small ratios the error is quite substantial. Thus, on an intuitive basis, one would expect little success in accurately recovering the shape of non-spherical surfaces from shading. To confirm this expectation, we applied the shape evaluation procedure outlined earlier to the recovery of elliptical and hyperbolic surfaces using the canonical examples of an ellipsoid and torus. In the first example, figure 11a, an image of an ellipsoid with axis ratios 100:70:35 was generated with its principal axes parallel to the coordinate axes. The tilt error distribution for this surface is shown in figure 12 as a surface map with the highest point corresponding to a tilt error of approximately 20%. The resulting distribution of surface normals is shown in figure 11b, and a view of the reconstructed ellipsoid in figure 11c. The results of the reconstruction experiment for this example are summarized in Table 2 (Appendix C).

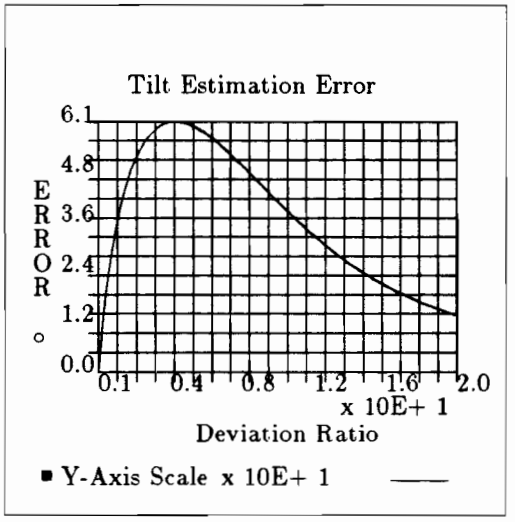

Figure 10 Tilt estimation error as a function of eccentricity

The basic result of these experiments, contrary to expectations and given the data in Table 2, is that relatively precise recovery of ellipsoidal surfaces is possible. On comparing the effects of different light source positions, it would appear that the L vector uncertainty, within the range specified earlier, results in a parameter variation of about 5%, similar to that

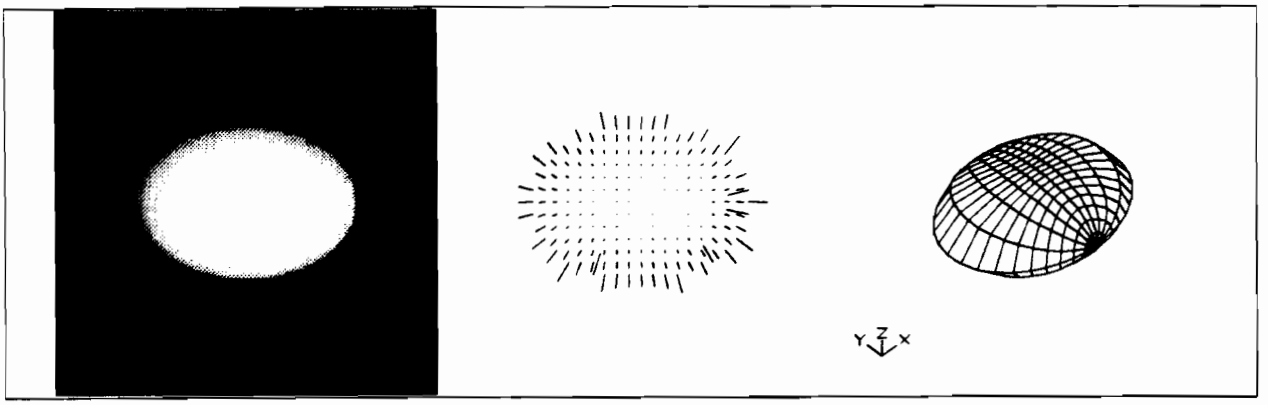

Figure 11a-11c Ellipsoid reconstruction: (a) original image (b) estimated surface normals (c) reconstructed surface

for a sphere. Note also that this variation also includes a component due to the reconstruction procedure, as the surfaces are randomly sampled to provide data for parameterization. All parameter values are within 5% of their actual values as recovered from an original image with a uniform error distribution of approximately 10%. What, then, accounts for the surprisingly good results? To answer this question we must consider the effect of tilt uncertainty on the surface integral (discussed in section 3.2), specifically on the height differential ΔZ.

If we consider the simplest form of interpolation on a surface, i.e. along a plane T_p tangent to a point p lying on a surface S at position (x_p, y_p) (figure 3), the height differential is given by (Appendix A)

$$\Delta z(\Delta x, \Delta y) = \frac{\Delta x \sin\sigma \cos\tau + \Delta y \sin\sigma \sin\tau}{-\cos\sigma}, \tag{31}$$

where σ and τ are the slant and tilt angles respectively at position (x_p, y_p) on surface S.

It is clear that errors in tilt estimation will ultimately affect the recovered shape of S through Δz, however, we must also take into account the effect of the $\sin\sigma$ term in (31). Figure 12 shows the distribution of τ error as a function of position on the ellipsoidal surface shown in figure 11a. Notice that where τ errors are large, σ angles are small. The effect of this rather fortuitous circumstance is that differential terms corresponding to large errors in τ are reduced, resulting in an attenuated reconstruction error. This is demonstrated in figure 13 which shows the distribution of differential error on the surface of figure 12. As it turns out, the

most significant errors occur on the periphery of the surface, which differs considerably from the tilt error distribution of figure 12. In addition, the magnitude of this error is sufficiently bounded so as not to introduce undue distortion into the surface reconstruction.

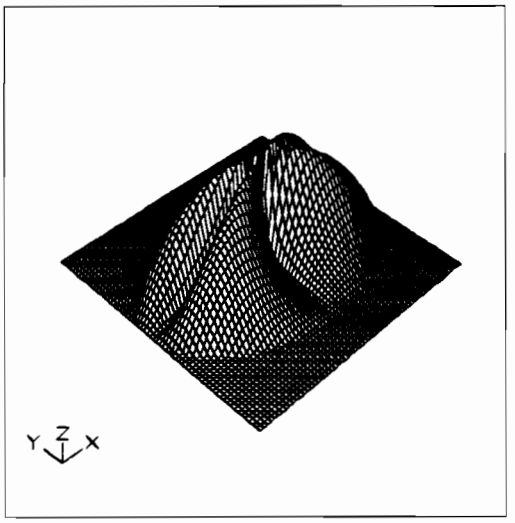

Figure 12 Tilt error distribution as a function of position on an ellipsoid

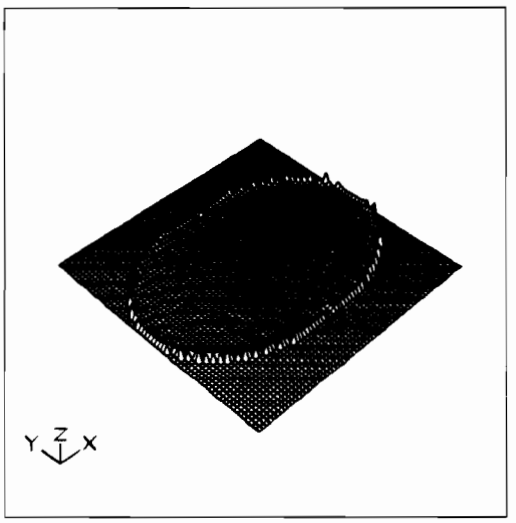

Figure 13 Height differential error as a function of position on an ellipsoid

Thus far our examples have dealt only with surfaces parallel to the XY plane having relatively low eccentricity ratios. A more realistic situation would be to consider high ratio surfaces which reflect either the projective foreshortening caused by rotating a surface into

the XY plane, or the generic structure of surfaces of high curvature. Our model in this experiment was an ellipsoidal surface centered at the origin with axes ratios of 100:20:50 as shown in figure 14a. The resulting surface normal field is shown in figure 14b and a view of the reconstructed ellipsoid in figure 14c. The latter is displayed along the Z axis which accentuates deformation caused by high eccentricity. The results are summarized in Table 3 (Appendix C).

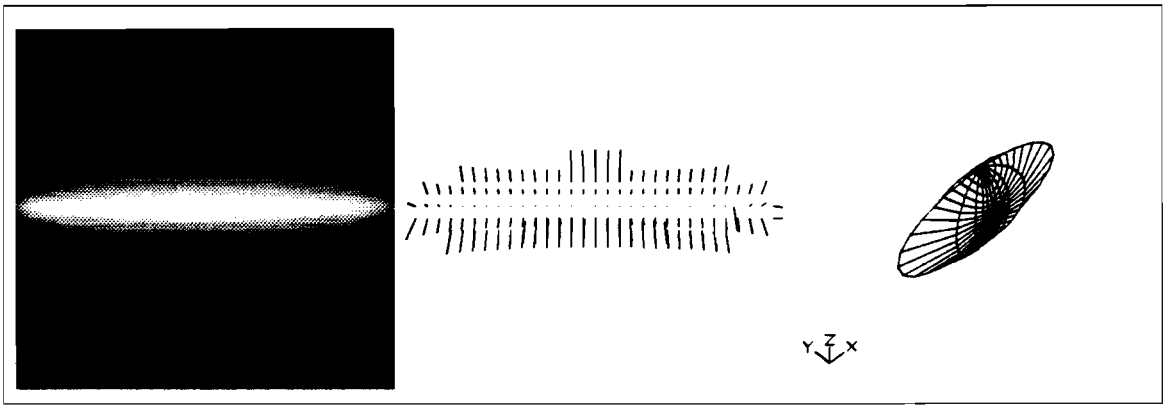

Figure 14a-14c Ellipsoid reconstruction (high eccentricity): (a) original image (b) estimated surface normals (c) reconstructed surface

As in the previous examples, light source position errors within the specified range of 20% solid angle, do not greatly affect the variation of recovered parameters. In general, rotation angles are more precise due to the elongation of the ellipsoid, which improves the estimation of axis directions. Axis ratios, however, are considerably more distorted, but still within 20% of their actual values. The observation here is that regions of high curvature on a surface, due either to foreshortening or actual structure, will be "stretched" out of shape. Shading information alone in these cases is insufficient for accurate surface recovery, however, it might still be sufficient for qualitative estimates of shape. To get an idea of the error component due to tilt estimation errors in this case, errors in the height differential, Δz, are plotted as a function of position on the ellipsoidal surface of figure 14a and shown in figure 15 to the same scale as in figure 14. As might be expected, there is a significant increase in the magnitude of the differential error along the periphery of the surface where slant angles are large.

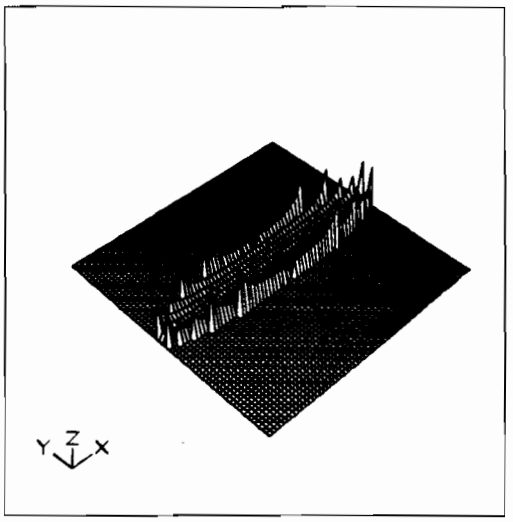

Figure 15 Height differential as a function of position on an ellipsoid with high eccentricity

As a final example, we consider the reconstruction of the half-torus shown in figure 16a. The criterion for evaluation used was the mean circularity of the circular cross-section, computed by radially sampling the torus and fitting an ellipse to each sample. We were not able to compute the error distribution of the source image[12], but estimate the mean error to be within 10% to 15% of the ideal intensity distribution. Figure 16b shows the resulting surface normal field, and 16c the reconstructed surface obtained from a single view. The results of this experiment are summarized in Table 4 (Appendix C).

For an ideal torus, the major axes of an ellipse fit to a cross-section should be equal, for all radial samples. Thus the mean value of cross-sectional axis ratios would seem to be a reasonable measure of shape fidelity for a torus. The results of the experiment indicate an acceptable reconstruction error, however it is not clear to what extent this is due to the eccentricity of the surface.

---

[12] The test image was not created by a generating function as with the previous cases, but constructed by rotating a circular cross-section. Earlier estimates were based on a comparison between the ideal intensity distribution provided by the generating function and actual points in the quantized image.

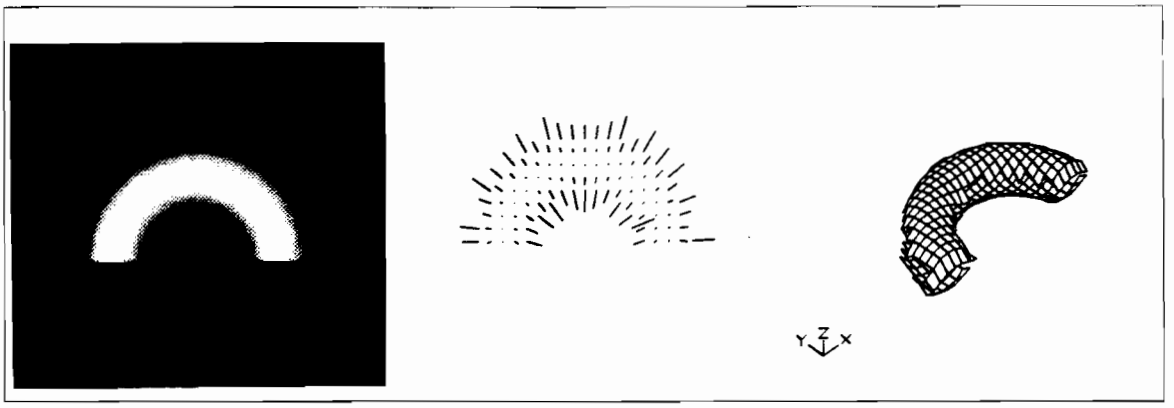

Figure 16a-16c Torus reconstruction: (a) original image (b) estimated surface normals (c) reconstructed surface

3.1.2.3 Localizing the Intensity Gradient

Up to now we have confined local surface estimation to smooth, convex surfaces. But what happens, for example, when we apply our estimation procedure to a non-convex surface such as one formed by the intersection of several convex surface components? An example is shown in figure 17a, the resulting surface normal field in figure 17b, and the reconstructed surface in figure 17c. The resulting normal field in the vicinity of the intersections is incorrect, resulting (in this particular case) in an inversion of what are supposed to be protrusions out of the surface (figure 17c). The reason for this error has to do with the locality of estimation regions[13] within an image, namely that they must not cross boundaries that correspond to physical discontinuities on a surface. Since surface orientation estimates must rely on support from continuous, local neighbourhoods, the accurate location of discontinuities within a neighbourhood is a prerequisite to surface recovery [Leclerc & Zucker 1984, Leclerc 1986]. In the above example, the estimation region for the intensity gradient direction was allowed to span discontinuities at the intersection of two surfaces. The result was that the estimated gradient directions had no validity in the vicinity of these intersections, leading to the results shown.

---

[13] The term estimation region refers to the neighbourhood of a point on a surface from which an estimate of local surface orientation is computed.

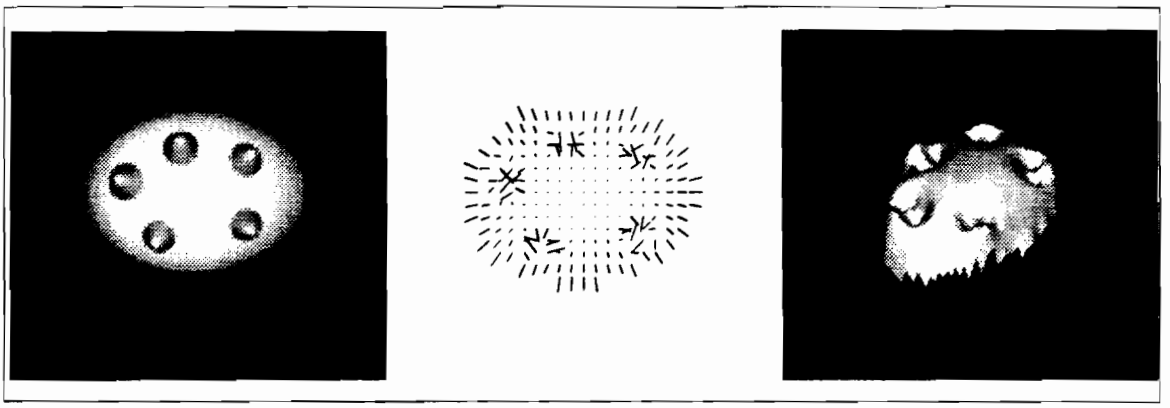

Figure 17a-17c Reconstruction of a non-convex object with incorrect estimation regions (a) original image (b) estimated surface normals (c) reconstructed surface

Thus, in order to estimate local surface orientation, it is important to first determine the location of discontinuities in the image intensity function, leading to a piecewise smooth decomposition of an image. Leclerc [1986] has addressed this problem in detail, observing that previous approaches to the problem were often based on inappropriate or incorrect models of edge discontinuities. His major points were that discontinuities cannot be accurately located without also determining the local structure of the underlying function, and that approaches based on a single model of edge discontinuity are bound to lead to incorrect results. His approach was to locally approximate the intensity distribution, adjusting the sampling functions such that they fulfilled conditions for local continuity. In this manner Leclerc was able to recover the structure of the intensity distribution, and in the process accurately locate discontinuities. With regard to local surface estimation, however, Leclerc's model also provides a local parametric description of the intensity function from which both the gradient direction and intensity value are immediately available.

Because the 2D version of Leclerc's procedure was not fully implemented at the time this research was conducted, we used a different approach. A Sobel operator was used to provide a first approximation to the possible locations of discontinuities in an image. Estimation regions were placed so as to lie to one side of an edge discontinuity, much in the same way that floor tiles line up against a wall (figure 18). The size of these regions was selected to be large

enough to account for errors in the placement of discontinuities. Unfortunately, there was no way to handle the false negative responses of the operator.

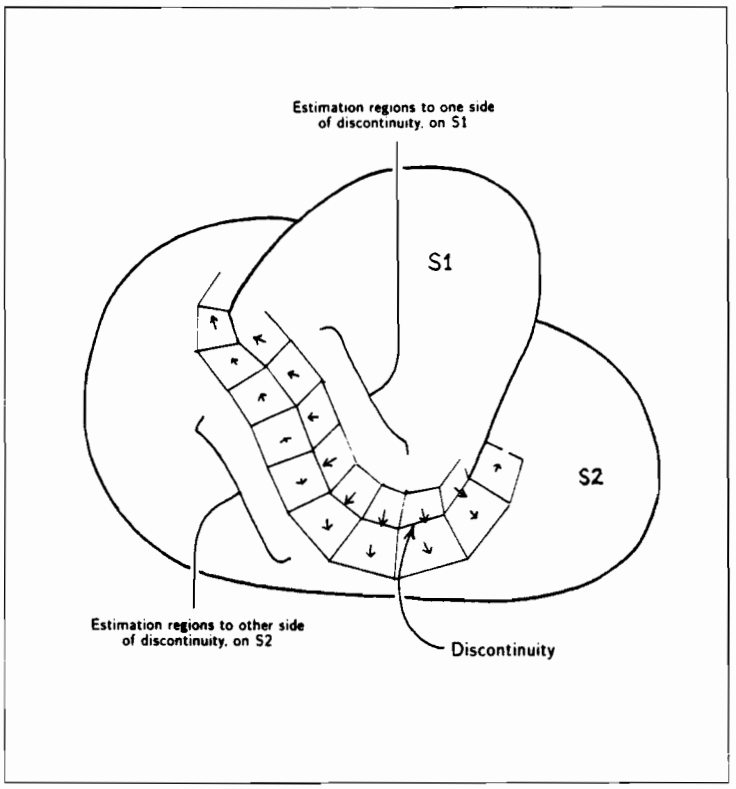

Figure 18 Placement of estimation regions

Once an estimation region is defined for a point, local surface orientation can be computed in a straightforward manner from measurements of intensity and its gradient direction. Intensity is obtained by smoothing the region with a Gaussian filter (figure 19a) and noting the intensity at the center of the neighbourhood. The gradient direction is estimated by convolving the region with a pair of directional derivative operators (figures 19b,c) and computing the arctangent of their magnitudes. The operator used was the directional derivative of the Gaussian filter:

$$f(x,y) = \frac{-x}{\eta^3\sqrt{2\pi}} e^{\frac{-x^2+y^2}{2\eta^2}} . \qquad (32)$$

The drawback to this method, however, is that the blurring induced by the Gaussian filtering tends to smooth out the surface whose orientation is being estimated, as well as the noise

component. It is, however, possible to devise operators that minimize the amount of blurring by making use of de-blurring operators [Kimia & Zucker 1983, Hummel, Kimia & Zucker 1984] to reduce the amount of distortion.

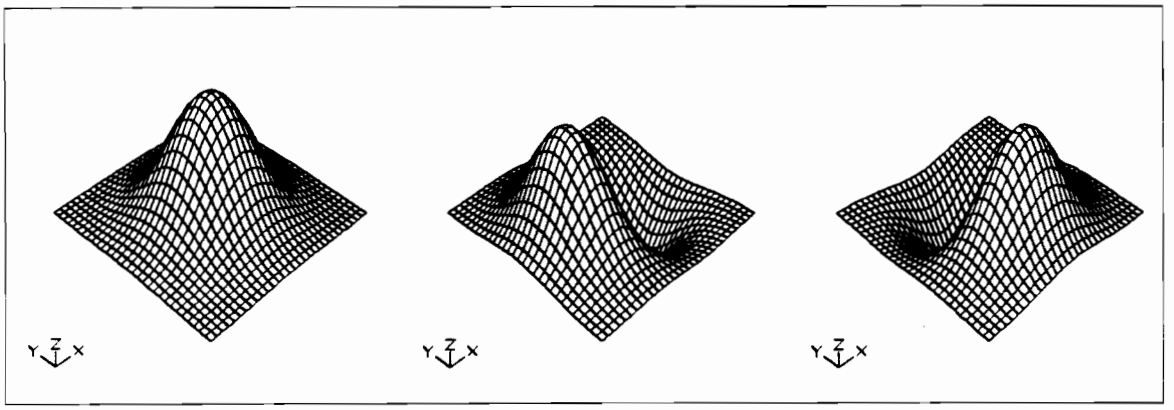

Figure 19a-19c (a) Gaussian filter (b-c) Directional derivative operators

The experiment on the image of figure 17a was repeated using the output of a Sobel edge detector as a guide for placing sampling neighbourhoods in estimating local surface orientation. The resulting surface normal field is shown in figure 20a and the reconstructed surface in 20b. Notice that orientation is now estimated correctly, resulting in a proper reconstruction[14] of the surface. This result is also interesting from the perspective of distortion in that the surface projections, which are Gaussian cones[15] of relatively high curvature, appear to have been recovered quite well. As it turns out, a Gaussian cone parallel to the XY plane satisfies (23) exactly.

3.1.3 Discussion

In this section we have seen that, under the appropriate constraints, a reasonable recovery of surface shape through shading is both possible and practicable. The major weakness in

[14] Since this surface is self-occluding, the location of occluding contours is also required in order to be able to reconstruct the surface. This topic is deferred to section 3.2.1.

[15] The surface function obtained by rotating a Gaussian profile about the axis of symmetry.

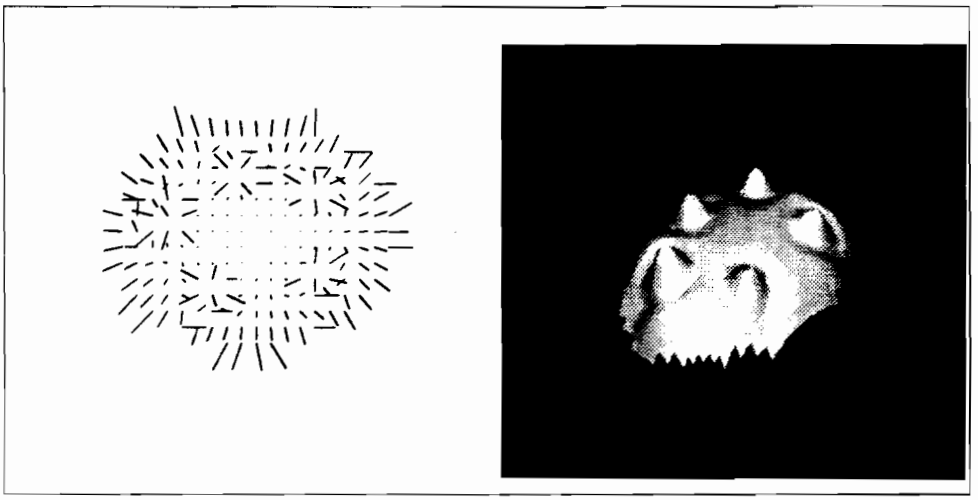

Figure 20a-20b Reconstruction of a non-convex object with correct estimation regions (a) estimated surface normals (b) reconstructed surface

the examples presented thus far, however, is that they all were of constant reflectance, ρ. A possible solution to this problem is presented in [Lee & Rosenfeld 1983], which outlines an iterative approach to estimating the $\rho\lambda$ term based on measurements of three neighbouring points along the direction of the intensity gradient.

A final note concerns the uniqueness of the resulting estimates and the precision to which fine curvature variations can be recovered. With respect to uniqueness, there are an infinite number of three-dimensional scenes which can produce the same two-dimensional image, given the projective nature of image formation. Consequently, it is not surprising that estimates of local surface orientation should be non-unique, therefore, how does our estimation procedure arrive at the "correct" interpretation? A possible explanation for this concerns the *scale* at which variations in surface curvature are recovered. For example, consider how the surface of the moon appears when viewing with the naked eye and through a powerful telescope. In the first case, the surface is predominantly smooth and spherical except for some of the larger craters that are visible to the unaided eye. However, in the second case, the magnified view provided by the telescope reveals considerably more detail of the topographic structure of the lunar surface.

What happens in the process of estimating local surface estimation is analogous to the moon surface example. For a given image resolution and operator size, there is an upper limit on the curvature variation (continuous) that can be recovered. High curvatures correspond to high spatial frequencies in the image domain, resulting in an upper bound on curvature variation for a given image resolution. Operators applied to the image further restrict what can be recovered as operator size is directly related to spatial frequency. So assuming that image resolution is sufficient to capture fine curvature variations, the scale (i.e. operator size) at which a computation is performed will affect the scale at which surface variations are recovered. How does one choose an appropriate scale? A complete answer to this question is not clear, however, our choice was governed by the *size* of the features that we wished to detect. For example, in the case of the object in figure 17a, the operator size was chosen to be small enough to provide an adequate number of samples in the region of a surface protrusion, yet large enough to be a valid approximation to the operation being performed.

3.1.4 Summary

Shape from shading techniques can be exploited in the recovery of local surface orientation provided that their limitations are clearly understood. We have shown that reasonable, qualitative shape estimates can be obtained in most circumstances where constraints are met, and in certain cases, albeit limited, quantitative shape recovery is also possible.

3.2 From Orientation to Depth

It is often more convenient to represent a surface as a function[16] of the form $z = f(x, y)$ (i.e. as a depth map) than as a field of surface normals. Besides being a more intuitive representation for surfaces, the functional form of a depth map is ideally suited to conventional

[16] We use the terms surface function and depth map interchangeably.

methods of approximation and analysis. However, the transition from orientation to depth is complicated by the need to determine boundary conditions and scale factors for integration, as well as by the quantization and noise error in the surface normal field. We begin by considering the surface integral in the ideal case of a smooth, non-occluded surface whose local orientation is known exactly at each point on the surface.

The depth $D(x,y)$ at any point p on a surface S is computed by integrating from a point p_o, whose depth is known, to p as follows:

$$D(x,y) = D(x_o,y_o) + k\int_{y_o}^{y}\int_{x_o}^{x} f(dx,dy) \tag{33}$$

where k is a scale factor for depth related to $f(dx,dy)$.

A suitable interpolation function for a dense orientation field is the tangent plane T_p to S at p. Substituting (31) into (33) we obtain

$$D(x,y) = D(x_o,y_o) + k\int_{y_o}^{y}\int_{x_o}^{x} \frac{\sin(\sigma_{x,y})\cos(\tau_{x,y})dx + \sin(\sigma_{x,y})\sin(\tau_{x,y})dy}{-\cos(\sigma_{x,y})}, \tag{34}$$

where $\sigma_{x,y}$ and $\tau_{x,y}$ are the slant and tilt angles at a point (x,y) respectively.

In general, surfaces tend to be at best piecewise smooth, and are often occluded by other portions of the surface, or other surfaces. Furthermore, estimates of local surface orientation may be inaccurate for the reasons cited in section 3.1, not to mention the effects of noise and quantization error. Thus, the surface reconstruction procedure implied by (34) must somehow take these effects into account in order to be of use in a more general class of surfaces. In the following sections, we discuss the reconstruction problem in view of discontinuities in the surface function and the minimization of reconstruction error.

3.2.1 Discontinuities in the Surface Function

The class of surfaces in which we are interested is assumed to be piecewise smooth, requiring a priori location of surface discontinuities and the determination of boundary conditions

for the integral (34). If the path of integration from a point p_o to p crosses a discontinuity on a surface, then the value of the surface function (i.e. depth) on the other side must be estimated or provided. The alternative is to integrate around the boundary and reach p by some other path, but this may not be possible. However, there is one constraint that we can take advantage of for a piecewise smooth surface. That is, depth changes across discontinuity boundaries *not* corresponding to occlusions are often smooth. This is because such events usually correspond to creases on a surface as opposed to depth discontinuities on "wedding cake" surfaces (figure 21), for example. It is thus very important not only to be able to locate surface discontinuities, but also to be able to differentiate between occluding boundaries and creases on a surface.

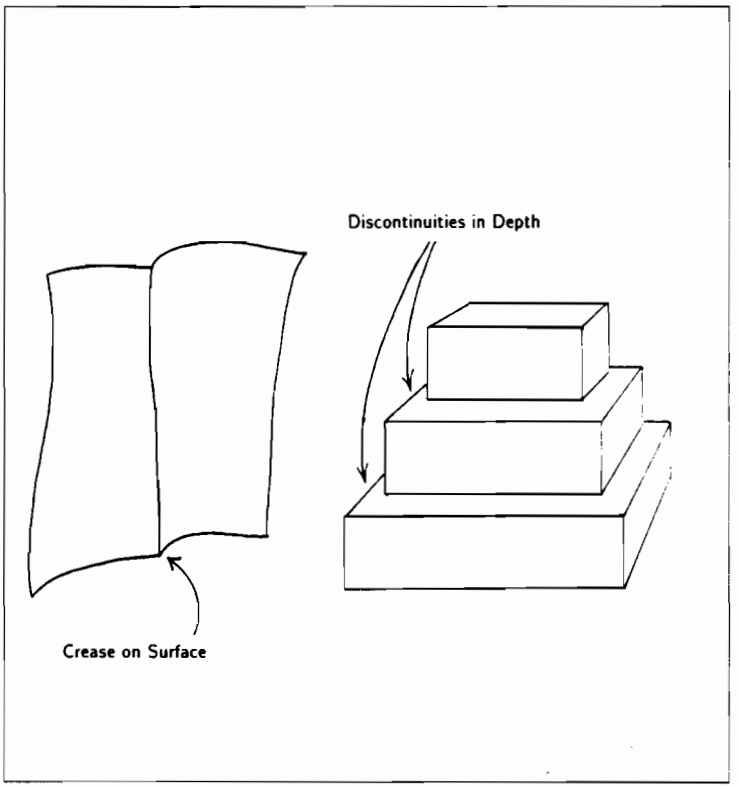

Figure 21 Different types of depth discontinuity

The location of discontinuities on a surface is often approached by looking for corresponding changes in image intensities [Leclerc & Zucker 1984, Marr 1982]. Image intensity values are a function of surface geometry, reflectance, scene illumination, and viewing position

[Marr 1982]. By invoking general position assumptions, i.e. that on average, different surface properties do not change simultaneously, an argument can be made for associating an image discontinuity with a single event. Unfortunately it is often difficult to determine the exact nature of a discontinuity, i.e. the physical change responsible. However, by examining the behavior of the intensity function in the neighbourhood of a discontinuity, i.e. what Leclerc calls *the local structure*, it is possible to gain some insight into the type of surface change associated with a particular image discontinuity [Leclerc & Zucker 1984; Leclerc 1986]. For example, a discontinuity due to a cast shadow is distinguishable by virtue of the fact that the profiles of the derivative of the intensity function on either side of the discontinuity are identical.

Another source of information about surface discontinuities is provided by the surface normal field itself. For example, at an occluding contour on a smooth surface, the Z or *slant* component of the normal rolls off to 90°. Occluding contours are important boundary conditions for the surface integral since depth information may not be propagated across them. We can, in fact, adopt an even more conservative strategy by treating *any* evidence for a discontinuity as a boundary around which to integrate or propagate depth information. This would include false positive indications often associated with discontinuity detection in image intensities. However, this poses little difficulty as long as a path exists that connects each point on a surface. This strategy might be viewed as a variation on Grimson's [1981] "No news is good news" theme in which lack of contrary evidence is seen as an indication of a smooth surface.

We can now summarize the integration procedure for a piecewise smooth surface in the following steps:

- Determine the set of possible discontinuities on the surface by computing the locations of discontinuities in the image intensity array.
- Identify points on the surface corresponding to occluding contours by examining the slant

component of the surface normal.

- Apply the integral (34) to the field of surface normals, selecting a path of integration that does not cross any possible discontinuity or occluding contour.
- Resolve any isolated regions on the surface by assuming that depth is continuous across a surface discontinuity not corresponding to an occluding contour.

3.2.2 Minimizing Reconstruction Error

The application of (34) to reconstructing the surface function is inappropriate in all but ideal circumstances where orientation is known exactly at each point on the surface. The problem that we are actually interested in solving is that of finding a surface function $z = f(x, y)$ that best fits orientation estimates. Ikeuchi [1983] used a least squares approach to solving the problem by seeking a function that minimized the following:

$$\int\int (z_x - p)^2 + (z_y - q)^2 dxdy, \tag{35}$$

where z_x and z_y are the first partial derivatives with respect to x and y of the surface function $f(x, y)$, and p and q, the x and y components of the estimated surface normal.

The solution to (35) is a problem of the calculus of variations [Courant & Hilbert 1953] from which a differential formulation of the problem is obtained by applying Euler's differential equation [Courant & Hilbert 1953; Ikeuchi 1983; Terzopoulos 1984],

$$\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} = p_x + q_y, \tag{36}$$

where p_x and q_y are estimates of the derivatives of p and q in the x and y directions.

Ikeuchi [1983] solved (36) with the following iterative procedure:

$$z^{n+1} = \bar{z}^n - \rho(p_x + q_y), \tag{37}$$

where $\bar{z}$ is a local average of the surface function at a point on the surface, and ρ is a weighting term proportional to the neighbourhood in which $\bar{z}$ is computed.

Ikeuchi's algorithm is implemented as a parallel iterative procedure on a discrete grid of orientation estimates such as provided by the local shading analysis described in section 3.1. Discontinuities in the surface function are handled in exactly the same manner as the surface integral (34), in that the propagation of depth values is governed by the same path rules (e.g. depth values cannot propagate across occluding contours). The principal drawback with this method, however, is that convergence tends to be slow, especially on a self-occluded surface. Ikeuchi's procedure can be augmented, however, to resolve this deficiency in one of two ways. The first method consists of computing a rough estimate to the surface function by applying the integral (34), followed by Ikeuchi's procedure for an iterative refinement. The second, and more elegant, solution is to adopt the approach of Terzopoulos [1984] and apply the procedure at different resolutions. Convergence is speeded up considerably because depth values computed at coarse scales can be used as a starting point for computation at finer resolutions.

Another approach to the minimization problem, that is preferable when estimates are more noisy, is based on the following statistical argument. The error in the depth estimate at a point p is dependent on the path of integration from a known point p_o to the unknown point p, since it is the sum of errors at each point comprising the interval. If the error is assumed to be normally distributed, and we approach p from a sufficient number of different paths, then the estimate to p, $\hat{p}$, should approach p in the mean. This suggests an algorithm (Appendix B) based on generating a pseudo-random path of integration from a known point p[17] that covers each point on the surface, and takes into account discontinuities on the surface. If this procedure is repeated for a sufficient number of iterations, and the resulting depth maps

---

[17] The point p can be set to an arbitrary value if the actual value is unknown, the result being a relative depth map.

averaged, a good approximation to the actual surface function should be obtained. In fact, this is the procedure that was applied in section 3.1 in the recovery of scene generation parameters from local shading analysis. Figure 22a shows a 128×128 image of a stone owl[18] taken with a CCD television camera, and figure 22b the field of surface normals obtained with the shape from shading procedure described in section 3.1. We applied the above reconstruction procedure and recovered the surface function shown in figure 22c.

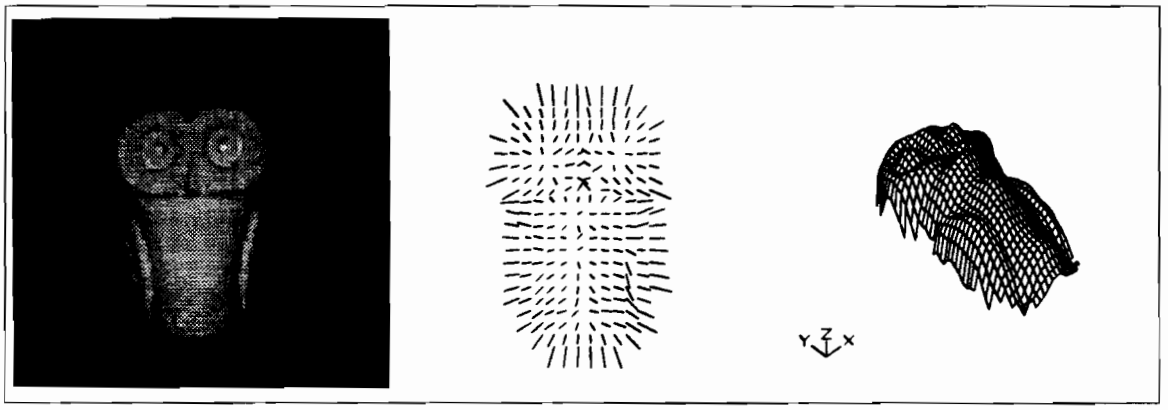

Figure 22a-22c Reconstruction of the owl (a) original image (b) Estimated surface normals (c) reconstructed surface

3.2.3 Summary

If estimates of local surface orientation are available for each point on a surface, then it is a relatively straightforward process to recover depth from orientation. Two problems, however, must be contended with: the localization of discontinuities on a surface, and the minimization of reconstruction error. Of the two, the former is the more difficult because of the ambiguity in associating image cues with physical events on a surface. The minimization problem can be handled with methods of variational calculus, although an argument was made for an alternate approach based on a statistical approach.

---

[18] The author wishes to thank his supervisor M.D. Levine for permitting his stone owl to be martyred for this project!

3.3 Surface Representation For a Single View

Our concern, up to this point, has been the problem of computing the set of points which define the visible surfaces of an object in 3D from information encoded in image intensities. We now turn our attention to the other major problem, that of computing a visible surface representation from this data. The principal representation for surfaces is the surface graph, G^c, that describes the surfaces of an object in terms of features of the differential geometry. The graph is compiled from a sequence of views of the object, of unknown relationship to each other, by which the complete surfaces of the object are presented to the viewer. We first consider how a surface representation is computed for a single view, then, in the following chapter, how this description is augmented with information from different views.

A subgraph of G^c is the single view surface graph G, which describes a surface in a viewer-centered coordinate system, i.e. in the coordinate frame of the surface function $z = f(x, y)$. The surface S is described in terms of a set of patches, S_i, which are further composed of a set of feature vectors, F_j. Processing proceeds from local to global by first computing F_j on a pointwise basis, and then using extrema in the F_j as a basis for decomposing S into S_i. The task of computing the set of F_j is equivalent to that of computing *Intrinsic Images* [Barrow & Tenenbaum 1978]. This is a set of registered arrays in a viewer-centered (retinotopic) coordinate system with each corresponding to a component of the feature vector F_j. We refer to this representation as a *Static Intrinsic Image* to denote the fact that it is associated with a single view of the object.

3.3.1 Computing the Static Intrinsic Image

A static intrinsic image consists of the following registered arrays, which contain the values of the corresponding features in image coordinates:

- Image intensity array.

- Orientation vector field of surface normals.
- Depth map (i.e. the surface function $z = f(x, y)$).
- Discontinuity map of image intensities.
- Principal curvatures of Π .
- Principal directions of Π .
- Loci of inflections in curvature.
- Loci of peaks in curvature.
- Intrinsic surface type labeling.
- Surface patch region map.

The first three items in the above list are *not* intrinsic features as they are dependent on viewer position, however, they are included for convenience in the representation. We have already discussed the problems of computing orientation and depth from image intensities in sections 3.1 and 3.2 respectively. The detection of discontinuities comprises an entire area of research on its own [Leclerc & Zucker 1984; Leclerc 1986], and we assume that this information is available through the application of appropriate methods. The remaining features are all functions of the second fundamental form of differential geometry, Π , and are discussed in the following sections.

3.3.1.1 Approximation

In order to compute Π , we require that the surface S be defined as a continuous function of order ≥ 2 of the form $z = f(x, y)$. What we have, however, is a set of points defining such a function in the form of a depth map. Moreover, we make the assumption that S can be piecewise continuously decomposed on the basis of discontinuities in the intensity array[19].

[19] In order to guarantee this assumption, we have assumed that surfaces are of constant reflectance, void of shadows, and illuminated by a point source at infinity.

Our approach to computing Π consists of approximating S in a local neighbourhood of a point p with a suitable function $\mathbf{f}$ from which Π can be computed by applying (1). For convenience, we can sample the set of points defining $z = f(x, y)$ in a rectangular neighbourhood about p, provided the neighbourhood does not lie in the vicinity of a discontinuity on S. In this case we alter the shape of the sample neighbourhood such that it lies to one side of the discontinuity.

The choice of $\mathbf{f}$ is dictated by consideration of the continuity requirements of Π , the expected characteristics of the surface function S, and sample variation in terms of noise and quantization error. In addition to a particular functional form, $\mathbf{f}$ has associated with it a scale dependent on the size of the sampling neighbourhood. This is analogous to the notion of scale in multi-resolution processing, with operators of different size corresponding to events at different spatial frequencies [Marr & Hildreth 1980; Witkin 1983]. Events corresponding to larger operators, which are presumably more stable given noise and quantization effects, are used to guide and evaluate the detection of events at finer scales [Zucker & Parent 1982,84; Leclerc & Zucker 1984, Leclerc 1986; Dill & Levine 1985; Terzopoulos 1984]. We considered only a single scale of approximation, with the sampling neighbourhood chosen so as to be responsive to the highest spatial frequency of interest. This has the effect of smoothing out surface variations sometimes associated with the texture of the surface, while serving as a good approximation to the more global geometry.

Once a scale of approximation has been selected, it remains to choose $\mathbf{f}$ keeping in mind the criteria stated above. In order to satisfy the continuity requirements of Π , $\mathbf{f}$ must be of at least second order. Furthermore, we have an additional constraint relating to shape from shading considerations [Pentland 1982a,b] which assumes S to be locally second order in the neighbourhood of p. It is also desirable to have some smoothing in the approximated surface to account for the effects of noise and quantization error. A function that satisfies these requirements is the following third order polynomial,

$$\mathbf{f} = a_1x^3 + a_2x^2 + a_3x + a_4x^2y^2 + a_5x^2y + a_6xy^2 + a_7xy + a_8y^3 + a_9y^2 + a_{10}y + a_{11}. \quad (38),$$

The choice of a polynomial approximation function was a matter of convenience since features such as (1) are easily computed in terms of polynomial coefficients. For most cases, polynomial fitting using least squares methods [Johnson & Wichern 1982] is of about the same order of computational complexity as convolution. This is because **f** is sampled in a coordinate system relative to p, meaning that the $(\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'$ term remains constant in (7). Polynomial fitting thus reduces to a matrix multiplication where the **A** term in (7) is dependent on the sample of S in the neighbourhood of p. The exception, of course, occurs in the neighbourhood of a discontinuity where the $(\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'$ term must be re-computed to account for sampling around discontinuities.

3.3.1.2 Differential Analysis

Once the coefficients of the approximating polynomial are determined, principal curvatures and directions are obtained through straightforward computation of the second fundamental form, $\mathrm{II} = \xi' A \xi$. Substituting the polynomial (38) into the expression for II (1), we obtain the following expression for A, evaluated at $p = (0,0)$:

$$A = \begin{pmatrix} \frac{2a_2}{\sqrt{1+a_3^2+a_{10}^2}} & \frac{a_7}{\sqrt{1+a_3^2+a_{10}^2}} \\ \frac{a_7}{\sqrt{1+a_3^2+a_{10}^2}} & \frac{2a_9}{\sqrt{1+a_3^2+a_{10}^2}} \end{pmatrix}. \tag{39}$$

The eigenvalues of A correspond to the principal curvatures, and the eigenvectors of A to the principal directions on surface S at a point p. In order to minimize the computation of principal curvatures and directions, it is convenient to solve (39) symbolically and obtain expressions in terms of the coefficients of (38) rather that attempt a numerical solution at each point on S. In this manner we obtain for the principal curvatures, κ_M, κ_m,

$$\kappa_M = -\sqrt{a_3^2 + a_{10}^2 + 1}\sqrt{a_9^2 - 2a_2a_9 + a_7^2 + a_2^2} + \sqrt{a_3^2 + a_{10}^2 + 1}\left(\frac{-a_9 - a_2}{a_3^2 + a_{10}^2 + 1}\right),$$

$$\kappa_m = \sqrt{a_3^2 + a_{10}^2 + 1}\sqrt{a_9^2 - 2a_2a_9 + a_7^2 + a_2^2} + \sqrt{a_3^2 + a_{10}^2 + 1}\left(\frac{a_9 + a_2}{a_3^2 + a_{10}^2 + 1}\right), \quad (40)$$

and for the principal directions, θ_M, θ_m,

$$\theta_M = \begin{pmatrix} 1 \\ -\frac{\sqrt{a_9^2 - 2a_2a_9 + a_7^2 + a_2^2} - a_9 + a_2}{a_7} \end{pmatrix},$$

$$\theta_m = \begin{pmatrix} 1 \\ \frac{\sqrt{a_9^2 - 2a_2a_9 + a_7^2 + a_2^2} + a_9 - a_2}{a_7} \end{pmatrix}. \quad (41)$$

We now move on to the problem of computing extrema in surface normal curvature, i.e. peaks and inflections in Π . Let $\kappa_M(x, y)$ define a grid with origin at p, and whose elements correspond to the maximum curvatures at the corresponding grid locations. Let $\kappa_m(x, y)$ define a similar grid, but corresponding to the minimum curvatures at each location. If a point p is a local maxima in Π , then p will be a local maxima in either $\kappa_M(x, y)$, $\kappa_m(x, y)$, or both. A suitable method of determining this analytically is to locally parameterize $\kappa_M(x, y)$ and $\kappa_m(x, y)$ using a suitable approximation function, and apply a positive definiteness test to the Hessian **H** [Strang 1980]. Using the same polynomial approximation as (38), the Hessian is given by

$$\mathbf{H} = \begin{pmatrix} 2a_2 & a_7 \\ a7 & 2a_9 \end{pmatrix}. \quad (42)$$

Inflections are identified as points through which the sign of Π changes for some particular direction θ on S. Using a similar approach to above, we could determine whether the approximation function is zero in the vicinity of p for either $\kappa_M(x, y)$ or $\kappa_m(x, y)$. Note that this is equivalent to performing the same operation on $K(x, y) = \kappa_M(x, y) * \kappa_m(x, y)$, the corresponding Gaussian curvatures. If S is defined by a sufficiently dense number of points, then a suitable approximation function **f** is that of a plane centered at p. The relative simplicity of a first order polynomial makes the task of computing its intersection with the surface S computationally efficient. This procedure is carried out for each point p on S, with inflection

points identified by the intersection of S and **f** occurring at less that half the distance between p and its immediate neighbours.

The effect of successive approximations to S is a smoothing of the original function. This has the desirable effect of minimizing the effects of noise, and the undesirable effect of distorting the actual form of the surface. Again, as was the case earlier in estimating shape from shading, the *scale* at which a computation is performed is the determining factor in the recovery of S or features of S. The approximation neighbourhood must be large enough to meet minimum sampling requirements with respect to least squares criteria, and small enough to capture variations at the highest spatial frequency of interest. We effectively "undo" some of the distortion introduced by smoothing at the geometric inference stage. When the local geometry of a surface is approximated by a volumetric primitive, parameterization is based on the initial estimates of S_i as opposed to the approximations on which features are computed.

The sign of Gaussian curvature also serves as a label for the *type* of surface lying under p as follows:

- (+) elliptical surface
- (0) parabolic surface
- (-) hyperbolic surface

Another useful piece of information that can be extracted is whether an elliptical point is convex or concave. This information is easily derived by looking at the signs of the principal curvatures with positive curvatures indicating a concave, and negative curvatures a convex surface at p on S.

3.3.2 Computing the Surface Graph

With the computation of the static intrinsic image, the surface S is represented on a pointwise basis in terms of the set of feature vectors, F_j, the most local representation

for surfaces in the shape hierarchy. The next step up in this hierarchy is a representation for surfaces as $\bigcup_i S_i$, where S_i defines a patch of S delimited by discontinuities on S and peaks or inflections in Π . In fact, we make a distinction between two kinds of patches. The first group, which we call peak patches, consists of surface regions delimited by peaks in curvature and discontinuities on S[21]. In the second group, called inflection patches, delimiting boundaries are made up of points corresponding to inflections in Π . Since these two groups are not mutually exclusive, a reasonable question arises as to how S is decomposed into a unique set of non-overlapping S_i.

Our solution was to represent S in terms of two decompositions based on the two categories of points. Each provides a decomposition of S which can be exploited in different ways by the process of geometric inference[22]. Surface patches are represented by two region maps[23] in the static intrinsic image corresponding to the two surface decompositions. Computing the patch region maps consists of the following steps: (1) the grouping of delimiter points into contours, and (2), the decomposition of S into sets of points isolated by these contours. The grouping problem that we are concerned with is best described by what Zucker [1982] calls the Type 1 grouping problem. This results in contours "that are highly accurate in position and hence permit large changes in orientation or curvature." A computational solution to this problem based on variational principles has been developed by Zucker & Parent [1982,84], the details of which are not discussed here.

The single view surface graph G is implicitly constructed by computing the set of intrinsic

---

[21] Discontinuities are peaks in the curvature of a surface. However, we make the distinction because the surface function is not defined at a discontinuity.

[22] The geometric inference paradigm presently considers only those patches delimited by peaks in the curvature function. These have a precise meaning in terms of the contours formed by the intersection of convex volumetric primitives.

[23] A region map is an array of labels in image coordinates where the label associated with a point p specifies the patch to which it belongs.

images describing the surface S. Patches corresponding to nodes of the graph appear as regions within a region map. Links are implicitly defined in terms of the adjacency between neighbouring regions. For a given point p on a patch S_i, the feature vector F_j corresponding to p is accessed by simply reading through the intrinsic images at the coordinates of p (figure 23). Thus, the static intrinsic image fulfills the requirements of the surface graph G as a representation for surfaces from a single viewpoint. We close this section with some example intrinsic images computed from the owl image of figure 22 to give the reader some familiarity with the features involved.

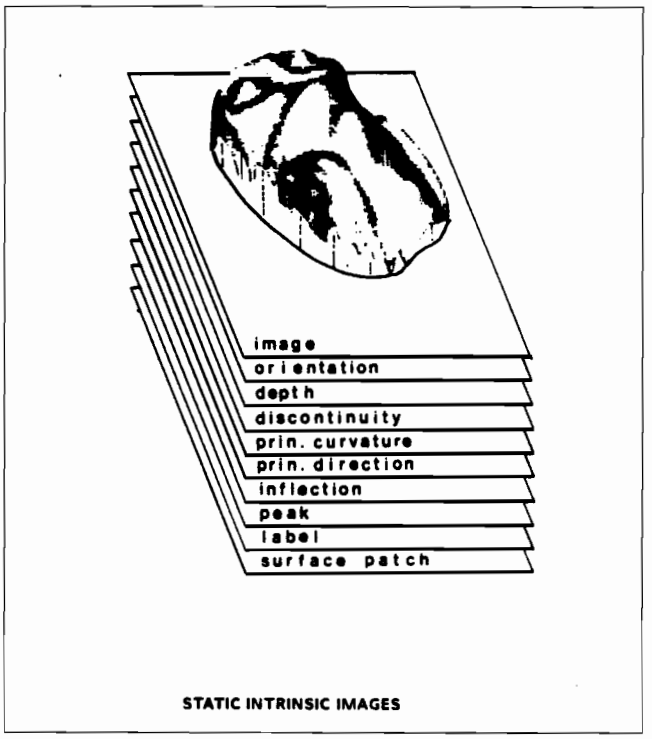

Figure 23 Static Intrinsic Images

Figures 24a and 24b show the distributions of principal maximum and minimum curvatures respectively on the surface of the owl shown earlier; brightness corresponds to curvature magnitude. The directions associated with maximum curvatures are shown in figure 25a with angle values corresponding to brightness. Contours formed by inflections in the curvature function are shown in figure 25b and those formed by peaks in figure 26a. Surface labels (i.e.

whether a point is elliptical, hyperbolic, or parabolic planar) appear in figure 26b overlaid on the original image. Elliptical regions on the surface are labeled as convex (-) or concave (+) and are shown in figure 27a. Finally, the decomposition of the surface into patches delimited by peaks in the curvature function is shown in figure 27b.

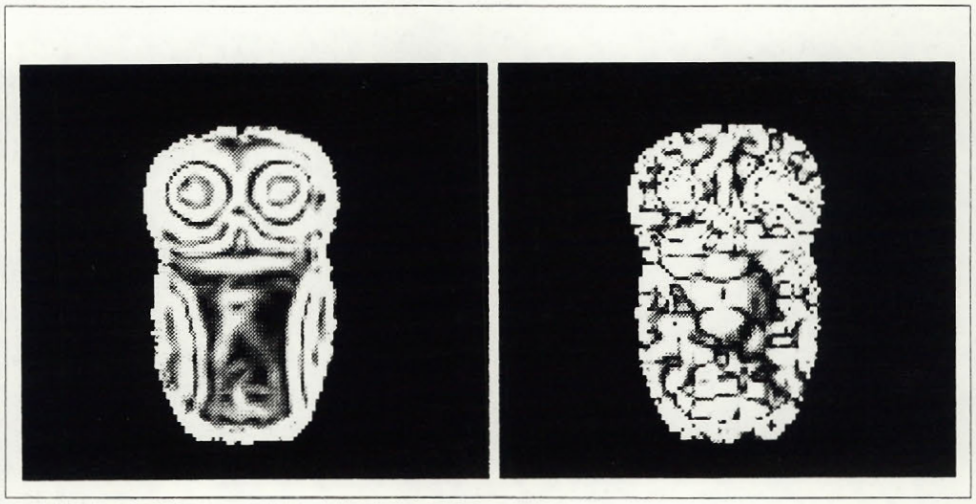

Figure 24a-24b (a) Owl principal maximum curvature distribution (b) Owl principal minimum curvature distribution

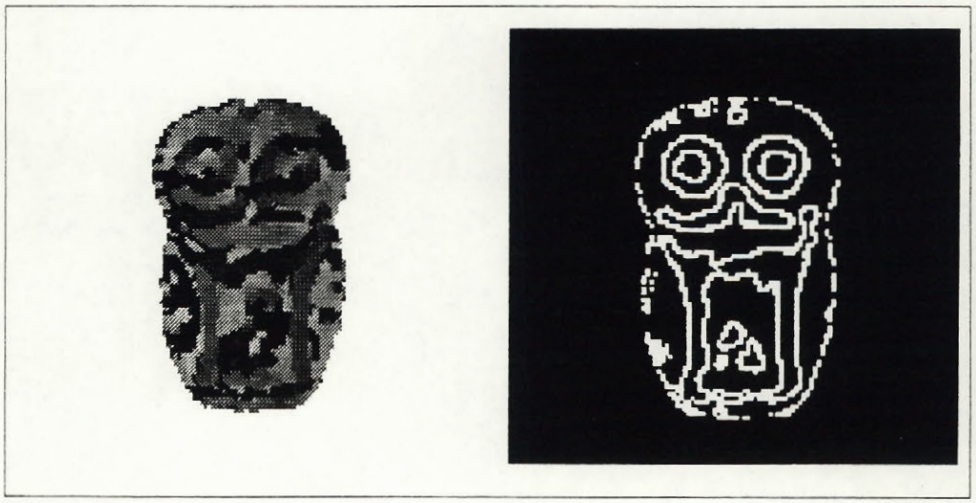

Figure 25a-25b (a) Directions associated with principal curvatures (b) Contours formed by points of inflection in the curvature function

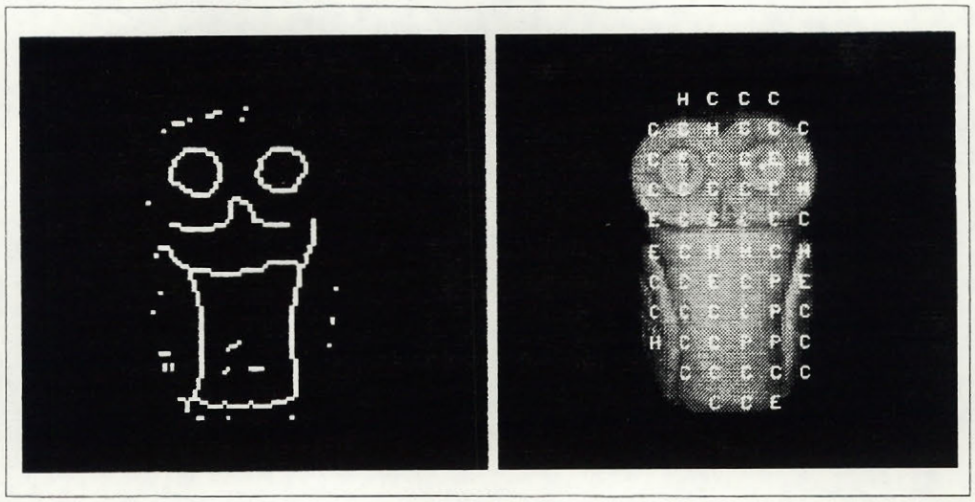

Figure 26a-26b (a) Contours formed by peaks in the curvature function (b) Surface labeling of elliptical, hyperbolic, and parabolic planar regions on the surface

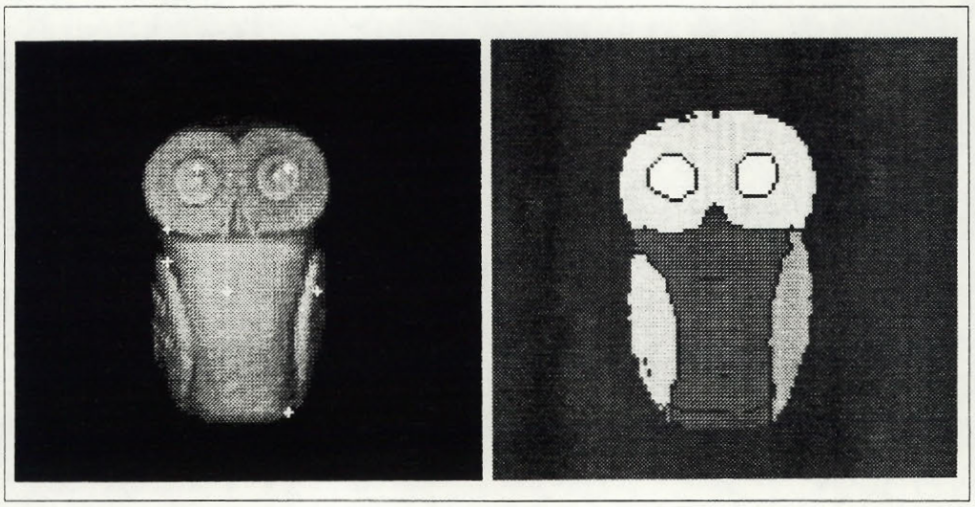

Figure 27a-27b (a) Labeling of elliptical regions as convex or concave (b) Decomposition of the surface into patches delimited by peaks in the curvature function

3.3.3 Summary

We have seen in this section how a description of a surface can be computed from a set of points in 3D that describe the surface in a viewer-centered coordinate frame. The computational methodologies of least-squares approximation and functional analysis were used to compute a description of the surface in terms of differential features and their extrema. A realization of the surface graph G was obtained as an equivalent set of intrinsic images which

are well-suited to computer vision applications.

3.4 Chapter Summary

The point of this chapter has been to examine the first of the three sub-problems that define the task of computing the 3D shape of a moving object. We have shown that in spite of fundamental limitations on the recovery of shape from shading, elliptical and hyperbolic surfaces can be reconstructed to an acceptable degree of accuracy. More general shapes can also be recovered, subject to distortion in areas of high curvature, but adequate for tasks of recognition. We also showed how a description of the recovered surfaces of an object could be computed using standard methods of approximation and functional analysis. The resulting description, the surface graph G, was implicitly computed in terms of a set of intrinsic images. Examples were shown of intrinsic images computed from data obtained with a low resolution, CCD television camera to demonstrate the practicability of the computational model in realistic task domains.

Chapter 4 Integrating Descriptions From Multiple Views

Unless an object possesses some degree of symmetry, a description obtained from a single viewpoint will not suffice as a representation for that object. For this reason we must consider how descriptions computed from several different viewpoints are integrated into a unique representation. Note that individual descriptions, such as the surface graph G, are tied to the frame of reference of the viewer. As was discussed in Chapter 2, we solve the problem of integration by choosing a common coordinate frame into which individual descriptions are mapped, upon computing the relationship between the common and viewer-centered frames. The result of this process, in the context of Chapter 2, is the composite surface graph, G^c, which provides a complete description of the surfaces of an object. In this chapter we will examine in detail the process by which G^c is obtained from a sequence of surface graphs computed from an object in motion about the viewer.

4.1 Estimating the Inter-Frame Transform

The first step in the process of computing G^c from a sequence of surface graphs, G_i, is finding the relationship between G_i and a suitable global frame of reference, W. Without any loss of generality, it is often convenient to choose W from among the frames corresponding to the G_i. Now if the inter-frame transform, T_i, can be determined which maps G_{i-1} into G_i for each frame in the sequence, then the composite transform relating G_i to W is obtained by

applying (3). Recall from Chapter 2 that T_i can be determined from a set of point correspondences between two surfaces, $S(i-1)$ and $S(i)$, represented by G_{i-1} and G_i respectively. The basis of our problem, then, is the task of computing these point correspondences. In the following sections we examine the problem of correspondence in terms of the computational solution proposed in Chapter 2.

4.1.1 The Correspondence Problem

The correspondence problem can be viewed as being comprised of the following elements:

- A set of tokens consisting of distinguished points on a surface.
- A metric for computing a measure of likeness, affinity [Ullman 1979], or compatibility [Hummel & Zucker 1983] between two tokens.
- A procedure for obtaining a consistent match between two sets of tokens.

The goal of any solution to the correspondence problem is to correctly match each token in one frame with its counterpart in another frame. The solution must also account for cases in which a token has no corresponding element, as might occur when a surface becomes occluded in a subsequent view.

Any point on a surface may serve as a token. However, points which are highly differentiated are better candidates since they are easier to identify uniquely. For this reason, we selected as tokens points at which surface normal curvature, Π , takes on extremal values, such as at peaks and inflection points. A description of a token point, p_j is given by its corresponding feature vector, F_j. A measure of affinity between two tokens may thus be computed in terms of a measure of the difference between the corresponding feature vectors. However, this is still an unsatisfactory solution to the problem in all but ideal cases due to the presence of noise and the effect of occlusion. Although F_j is a function of the local neighbourhood

of p_j, it is not a *stable* description when the local neighbourhood of p_j is encroached by an occluding contour, or when it is distorted by noise such as a specularity on the surface.

We can achieve a greater measure of stability in our description of p_j by extending the description of the token to include *the structure of the local neighbourhood* of p_j. In other words, rather than describing p_j in terms of a single F_j, we describe the token in terms of an $m \times n$ feature field **F** centered at p_j. A measure of affinity between two tokens can now be expressed in terms of a correlation of the structure of their local neighbourhoods using the functional (8). Consequently, tokens whose affinity is computed in this manner are less likely to get "lost" or incorrectly matched in the presence of noise than their single feature vector counterparts. More important, however, is the fact that in those circumstances where correspondence is in error, the more detailed structure of the local neighbourhood provides a means of determining the validity of a match. In particular, we may define a measure of confidence, C_i, associated with a match between $p_j(i-1)$ and $p_j(i)$, based on the value returned by the functional (8).

Besides relying on some threshold of C_i as an indicator of an invalid match, we may also exploit a more syntactic means of detection. This involves a comparison of the feature labels associated with a pair of matched tokens. The sign of Gaussian curvature, as well the signs of the principal curvatures in the case of elliptical surfaces, must be in agreement between two matching tokens at p_j. Note that for the purposes of vetoing a match, only those labels associated with the center of the neighbourhood, i.e. p_j, are examined. By applying these tests in addition to a threshold on C_i, we attempt to maximize the probability of obtaining correct correspondences where they exist.

The final element of the correspondence problem is the procedure by which we obtain a consistent set of matches between the two sets of tokens. Consistency, in this context, refers to a one-to-one mapping between the two token sets, unless, of course, corresponding elements are not present because of occlusion. Since one point cannot be in two places at the same

time, we cannot have many-to-one or one-to-many mappings in which points combine or split up (i.e. we are concerned with a subset of the more general problem addressed by Ullman [1979]). Given these constraints, we can formulate the desired procedure as a *consistent labeling problem* [Hummel & Zucker 1983]. Let the set of tokens to be matched correspond to the nodes of a graph Γ, with the spatial relationship between nodes described by arcs of Γ. To each node i of Γ, we assign a set of labels, Λ_i, that correspond to all possible candidate tokens in the succeeding frame. The labeling problem can thus be described as follows: to obtain for each node i of Γ a labeling that is consistent with constraints defined over the set of tokens.

A solution to the labeling problem can be found in a class of algorithms referred to as relaxation labeling processes [Hummel & Zucker 1983]. Let $p_i(\lambda)$ be defined as the probability that node i has label λ, such that

$$\sum_{j=1}^{m} p_i(\lambda_j) = 1.0. \tag{43}$$

Constraints are introduced to the labeling problem by functions $R_{ij}(\lambda, \lambda')$, which describe a measure of compatibility between nodes i and j for labelings λ and λ' respectively. These are in effect penalty functions by which local constraints can be expressed. For example, consider the paths of two neighbouring points on a rotating rigid body, where both points are assumed to lie to one side of the center of rotation (figure 28). One would expect these paths to be parallel for most neighbouring points on the surface, and this can be reflected in a compatibility function by penalizing labelings that correspond to points whose paths cross. The one-to-one mapping constraint can be similarly introduced by penalizing common labelings at two adjacent nodes. One can continue on in this manner and derive compatibility functions that reflect the constraints of a particular labeling problem.

With suitable compatibility functions defined, a measure of support for a particular label, λ, at node i, $\Sigma_i(\lambda)$, may be computed according to the following definition [Hummel & Zucker

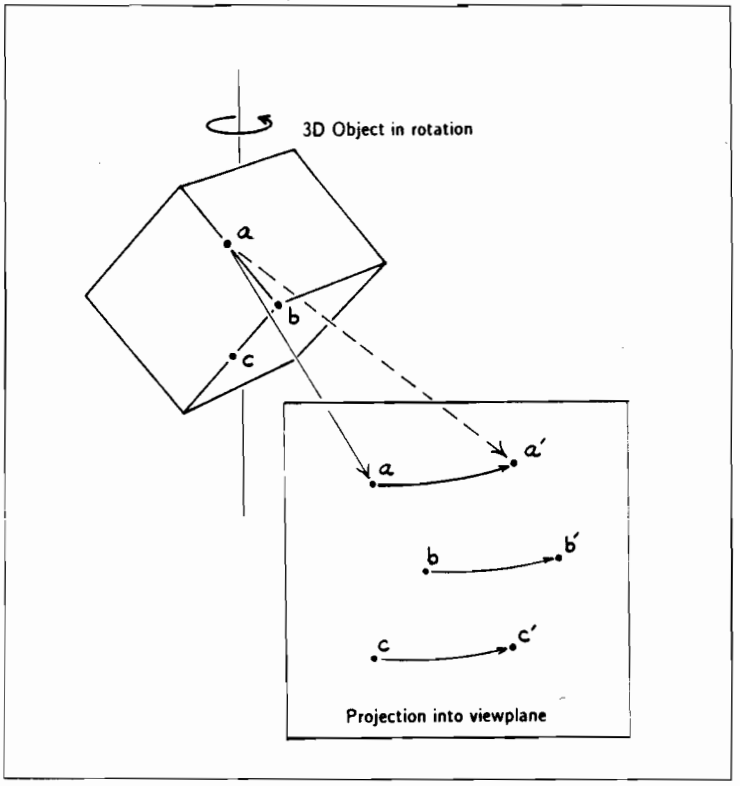

Figure 28 Parallel paths on a rotating rigid body

1983]:

$$\Sigma_i(\lambda) = \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} R_{ij}(\lambda, \lambda') p_j(\lambda'), \tag{44}$$

where j varies over the set of n nodes[24] of Γ, i excluded, and λ' varies over the set of m possible labels at each node i.

The support for each label λ_j at node i of Γ defines an m component vector, $\vec{p}(i)$. This vector is the basis by which the set of label probabilities for each node i, $p_i(\lambda_j), j = 1, m$, is updated at each stage of the relaxation labeling procedure. The updating strategy can be likened to a gradient search in an m-dimensional hyperspace Ω, where the support vector $\vec{p}(i)$ indicates the direction of travel, subject to the constraint defined by (43). The procedure is allowed to iterate until the respective label certainties converge, at which time the label with maximum

[24] In fact, j usually varies over a subset of Γ that defines the local neighbourhood of node i, since compatibilities with points outside of the local neighbourhood are zero.

certainty is selected for each node. A complete mathematical treatment of relaxation labeling processes and their relationship to other optimization methods is contained in Hummel & Zucker [1983].

4.1.1.1 Simplifying Constraints

The correspondence problem as formulated in terms of relaxation labeling processes can become computationally intractable as the number of nodes and labels in the relaxation graph grows large [Zucker 1978,81]. However, there are two constraints that can be used to simplify the problem:

- It is not necessary to match all the nodes of Γ.
- Some inconsistency in node labelings can be tolerated.

Both of these constraints are a result of how point correspondences are used in estimating the interframe transform T_i. Recall that two sets of point correspondences for a rigid body have the linear relationship given by (4). In order to estimate T_i it is sufficient to match only a *subset* of Γ, provided that the $\mathbf{B}'\mathbf{B}$ term in the linear regression (7) is well conditioned [Strang 1980]. Another consequence of the linear regression model is that we can tolerate some inconsistency in node labelings (i.e. incorrect matches) as long as the point correspondences are correct on the whole.

Application of the above constraints reduces the problem from finding a consistent labeling for Γ to that of a finding partially consistent labeling for Γ', where $\Gamma' \in \Gamma$. This is indeed a simpler problem than the one originally considered, but is still complicated by the fact that Γ' is not known a priori. A reasonable strategy to apply would be to search each node of Γ for the best matching label and then accept only those matches with a high measure of confidence as determined by (8). This labeling task is simplified if the confidence values associated with each label are strongly differentiated, and is precisely the motivation behind our use of

local neighbourhood structure in computing match affinities[25]. In this manner we attempt to maximize the difference between matching label affinities such that an unambiguous choice can be made. In the case where a match is ambiguous, we simply ignore it in the hope that a sufficient number of tokens can be matched with high confidence.

4.1.1.2 Pruning the Search Space

An examination of functional (8) shows that the task of finding correspondences for each node of Γ involves a search in a three-dimensional space defined by (x', y', θ'). In fact, the space would actually be larger if one took into account the deformation of the local neighbourhood of the token due to foreshortening. Fortunately there are constraints that can be used to resolve the problems of foreshortening and search space minimization. The constraints employed are:

- Locality of motion
- Rigidity of surfaces (locally in time)
- The local structure of objects in the domain

The most important of these constraints is the locality of motion which requires that matching tokens be in bounded proximity to each other. This is ensured by placing an upper bound on the velocity of objects during the interframe interval. There are two consequences to this constraint: the (x', y') region to be searched is restricted and the amount of foreshortening is limited so as not to be a problem in most cases. The result of severe deformations caused by foreshortening or occlusion will be a failure to find corresponding tokens. Recall that this does not pose a severe problem as we are required to match only a subset of Γ in order to estimate T_i. A general position argument can be made that in most cases not all of the visible surfaces of an object will be occluded or severely foreshortened.

---

[25] In the context of [Zucker & Hummel 1983], we are decomposing the relaxation labeling problem such that it is equivalent to local maxima selection.

We further eliminate candidates from consideration by noting that surfaces are assumed to be temporally rigid, i.e. that they do not deform to an appreciable extent during the interframe interval and that surface properties are maintained across corresponding tokens. In our case tokens are selected on the basis of extremal values in surface normal curvature. Candidate tokens without the requisite properties can be rapidly eliminated by simple comparison.

At this point we have restricted the search space of the correspondence problem to a subset of an x' by y' patch of surface at some unknown orientation θ'. Rather than search the complete space of orientations, we can exploit a further constraint on the local structure of surfaces. The extremal feature points that we use as tokens correspond to peaks or inflections in surface normal curvature and rarely occur as isolated points on the surface. Within a $u \times v$ neighbourhood of a token, we define *orientation* as the major axis of the point cluster formed by the token and its neighbouring extremal feature points (figure 29). This axis is determined by computing the eigenvectors of the two-dimensional inertia tensor matrix **I** corresponding to the projections of cluster points in the view plane. We define the eigenvector with the maximum associated eigenvalue as the major axis of the point cluster.

With the orientation determined it is straightforward to apply the correlation functional (8) and obtain affinity measures for each possible label of a particular node in Γ. Note that there is a 180° ambiguity in this orientation that makes it necessary to apply the functional in each direction. Another approach that avoids the foreshortening problem would be to compute the position and orientation of the point clusters in three dimensions using (12). In this manner the two clusters corresponding to a node and a candidate could be brought into alignment and the functional (8) applied. The only drawback to this method (and the reason we avoided it) is the computational complexity involved (i.e. computing the ICS of each cluster, determining the 3D transformation that maps the coordinates of one cluster into another, and coping with a 180° ambiguity in three axes instead of one.

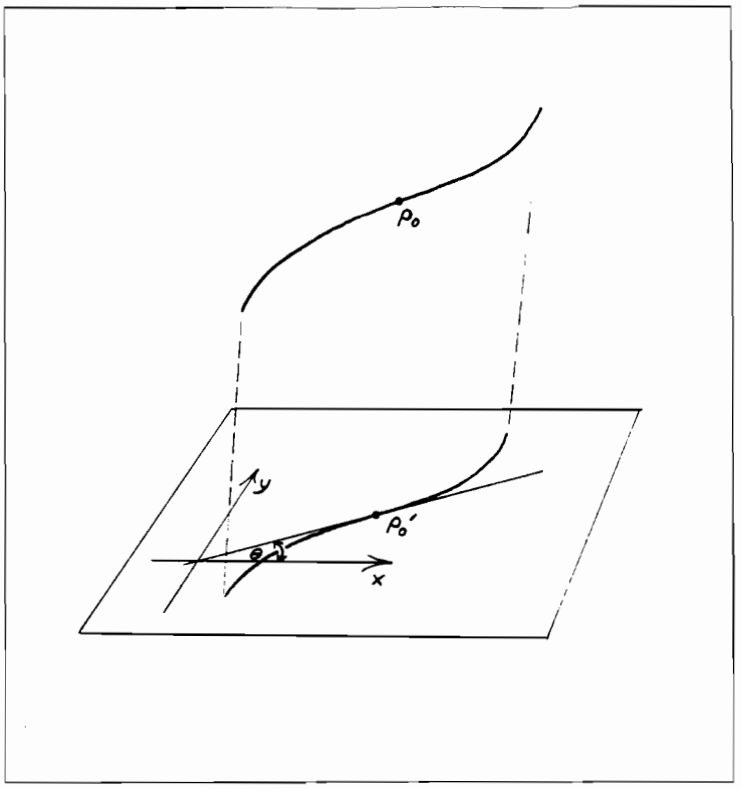

Figure 29 Orientation of the local neighbourhood of a token point

4.1.2 Experiments on Artificial Range and Image Sequences

Having defined the correspondence problem and its components (section 4.1.1), we look at the application of these concepts to solving the problem of estimating the motion of an object through a sequence of image frames. Specifically, we compute the set of interframe transforms, $T_i, i = 1, n$, that describes the motion by computing the correspondence between adjacent frames. Recall that we obtain an estimate of the interframe transform, $\hat{T}_i$, by applying the linear regression model (5) to the two sets of corresponding tokens under the assumption of rigid body motion. In the following sections we discuss some details of the implementation of our experiments, elaborating on the token selection process as well as how estimates of T_i are evaluated against their actual values. The major emphasis is on two sets of experiments that were designed to permit a detailed analysis of the correspondence process in terms of critical parameters and features.

4.1.2.1 Matching Algorithm

The problem of token selection is resolved by initially considering only those points on a surface associated with extremal features that mark the locations of change on the surface. These will presumably be more stable than randomly selected tokens, provided that they do not lie in the vicinity of occluding contours, and are readily identifiable as components of feature vectors F_j of surface S. We thus scan S for feature vectors meeting the stated requirements, and enumerate the corresponding surface points in the candidate set $L(F_j)$. Random selection is applied to $L(F_j)$ in cases where the number of selected tokens is large. We restrict the number of tokens[26] to ease the computational burden, yet ensure an adequate sampling of the surface. Where $L(F_j)$ is sparse, the procedure defaults to a random selection of surface locations associated with non-extremal features to make up the difference. This ensures that the subsequent estimation of T_i is based on a sufficiently large sample of S.

Upon selection of the set of candidate tokens $L(F_j)$ from frame $S(i-1)$ [27], correspondence is computed in the following manner for each token in the set:

- Define a $u \times v$ window in frame $S(i)$ centered at the coordinates of the token in frame $S(i-1)$.
- Enumerate a set of corresponding candidate tokens, $L'(F'_j)$, by searching the window for tokens whose corresponding F'_j has the same extremal property as the F_j of the token being matched.
- For each element of $L'(F'_j)$, compute the associated match affinity by applying the correlation functional (8).

---

[26] To about 400 tokens typically.

[27] The term frame is generally used in the context of correspondence in image coordinates. We refer to correspondence across frames and surfaces interchangeably because the surfaces that we are dealing with are described in a image coordinate system. For notational clarity, $S(i)$ refers to surface S contained in frame i, and S(i-1) as surface S in the previous or $(i-1)$th frame.

- Define two affinity thresholds, A_α and A_β, that correspond to the affinity of the best matching token and the affinity difference between the best and second best matching tokens respectively. If both thresholds are exceeded, and the matching token does not lie in the vicinity of an occluding contour, a correspondence is deemed to exist between the token in frame $S(i)$ and the token of highest affinity in $L'(F'_j)$.

The window size is chosen to encompass the probable movement of the token during the interframe interval as well as include the local neighbourhood of the token, and is determined by experiment. In a later section we describe the effects of local neighbourhood size in the match procedure.

In order to evaluate our algorithm for correspondence and the subsequent estimates of the interframe transforms T_i, we created two sequences of images such that the motion parameters were known exactly. This brings up the question of how we compare estimates of these parameters against their actual values. A simple parameter by parameter comparison does not give much insight to the "goodness" of our estimates, particularly since the resulting transformation matrices may not decompose nicely into rotations and translations. Our solution was to define the following error norm η:

$$\eta = \frac{1}{N_{(x,y,z)}} \sum_{(x,y,z)} \left\| T_i(x,y,z) - \hat{T}_i(x,y,z) \right\| \tag{45}$$

where $N_{(x,y,z)} \in S$ is the total number of tokens to which the transformations are applied.

The norm defines what we call a *mean displacement error*, that is, the average distance that a point is displaced from where it is supposed to be upon application of a known transformation. We find such a measure more useful[28] than specifying errors in translation and rotation because we are eventually confronted with a reconstruction problem in which pixel displacement errors are more intuitive.

[28] The effects of small rotations and translations are approximately the same.

Our interest, however, was not only in determining how well $\hat{T}_i$ represented the actual motion of an object, but also in quantifying different components of the estimation error. Errors are attributable to two main sources: (1) in computing and localizing features that serve as tokens, and (2) in the reconstruction of S upon which the former computations are based. We performed two sets of experiments in order to gauge these error components, the first on a sequence of range images of the model shown in figures 30a-30d, and the second on intensity images of the same model. By having direct access to range data, we bypass the reconstruction problem and can evaluate the correspondence process independently. Performing the experiment again on intensity images allows for a comparison of the two results, and the isolation of the reconstruction error component.

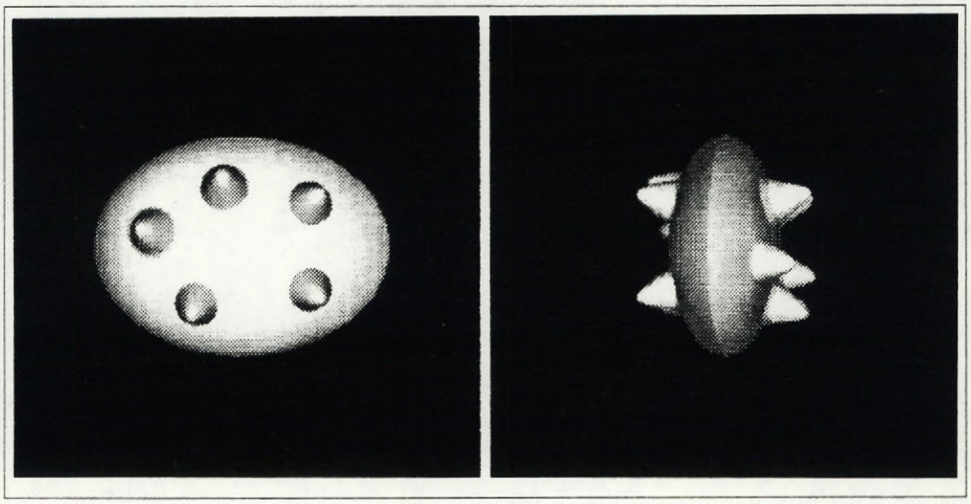

Figure 30a & 30b Artificial blood platelet model - 0 & 96 degree views

The correspondence computation is dependent on a number of factors which ultimately affect the estimation of the interframe transform T_i. In particular we consider:

- The feature weighting w_n in the correlation functional (8), i.e. the features used for comparison.
- The number of samples used in the least squares estimate of T_i.
- The procedure by which tokens are selected.

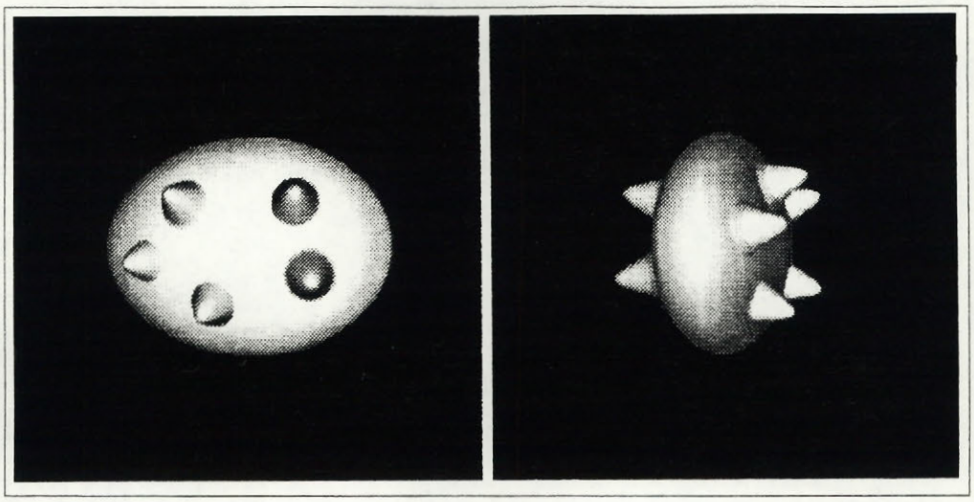

Figure 30c & 30d Artificial blood platelet model - 192 & 288 degree views

- The size of the local neighbourhood used as the token description.

The point of the two sets of experiments is to determine how each of the above factors influences the process and to compare these observations against expected behaviour.

The primary features used in (8) were the magnitudes of the principal curvatures and the intensity gradient[29] at points on the surface. We did not use the directions associated with principal curvatures because we found that their values had a high variance. This is to be expected given the smooth surfaces of the model used in the experiments. One interest in these experiments was a comparison between feature descriptors based on surface properties and those derived directly from the image such as the intensity gradient. Most approaches to point correspondence are based on features computed from intensity [Nagel 1978]. Because we are dealing with smooth surfaces however, features such as the intensity gradient are difficult to localize. Thus one would expect to do better with features directly tied to the surface. On the other hand, surface distortion in the reconstruction process will adversely affect the localization of surface features as well. For these reasons we applied weightings

---

[29] To avoid later confusion, note that the intensity gradient is computed on intensity images for experiments on both range and intensity data.

to (8) that emphasized curvature, the intensity gradient, and the combination of both in an attempt to gain some insight into feature selection.

The number of sample tokens used in correspondence is important because it affects the least squares estimate of the interframe transform T_i. We performed experiments in which sample size was varied in order to determine a lower bound on the number of samples required. In practice, we found that a sample size of 100 was more than sufficient to ensure a reasonable estimate of T_i for the point correspondences obtained with our procedure. Where the number of available tokens was less than this minimum size threshold, the remainder were made up with tokens randomly selected from the surface[30].

The manner in which tokens is selected is also important. Clearly, we wish to select those that are highly differentiated so as to maximize the difference in affinity measures between match candidates. For this reason we choose as tokens points that correspond to locations where surface normal curvature takes on extremal values. However, recall from Chapter 3 that these can also correspond to regions on the surface where reconstruction errors are likely to occur. We performed two sets of experiments in which tokens were selected randomly from points on the surface. The general expectation is that better results would be obtained with the more highly differentiated tokens. In cases of reconstruction error, though, the random selection approach might be one way of of distributing the risk to all parts of the surface, perhaps minimizing correspondence errors in the process.

Match affinities are computed by applying (8) to the local neighbourhoods of a token point in frame i and the local neighbourhoods of candidate tokens in frame j. The size of the local neighbourhood must be large enough to provide a stable description of the surface in the vicinity of the token point, yet be small enough to limit computation to a reasonable level. Noise in the surface reconstruction usually determines neighbourhood size, but another

[30] The procedure ensured that randomly selected tokens did not include those already selected or those lying in the vicinity of an occluding contour.

constraint is imposed by the rate at which the scene is changing. Neighbourhood size must be selected such that the entire region is visible in two adjacent frames for most of the tokens on the surface. In addition, foreshortening must also be considered as its effects are proportional to the size of the region involved. In our experiments, we considered two neighbourhood sizes and expected better results with the larger of the two.

Another aspect of these experiments concerns the detection of loss of correspondence, i.e. that point at which confidence in the estimate of T_i drops below an acceptable level. In order to force such events, we designed our object model such that correspondence could be made to degrade predictably as a function of rotation. The model used in these experiments is elliptical in its basic structure, such that it has a maximum and minimum area projection onto the viewing plane. In addition, surface protrusions are placed so as to maximally occlude each other when viewed from the angles corresponding to minimal area projection. The effect of this geometry is a cyclic variation in feature density that is designed to cause a smooth degradation resulting in the eventual breakdown of correspondence. From this behavior we can observe the effects of varying different parameters in optimizing correspondence, as well as the ability to detect a breakdown when it occurs.

4.1.2.2 Range Sequence Results

Figures 31a-31f contain plots of the mean displacement error, η [31], for six experiments on a sequence of range images produced from the artificial model shown in figures 30a-30d respectively. The actual sequence consisted of 30 frames (taken 12° apart) of the model in rotation, with a noise component that varied between 3% and 10% . Displacement error was determined by computing correspondence for each pair of adjacent images in the sequence, computing $\hat{T}_i$ from the corresponding tokens, and applying the error functional for η (45).

---

[31] The units of mean displacement error are pixels since the coordinate system of the reconstructed surfaces takes its dimensions from those of the input image sequence.

The data to which the functional was applied consisted of random samples of points lying on the model surface; T_i was obtained directly from the procedure used to generate the range sequence.

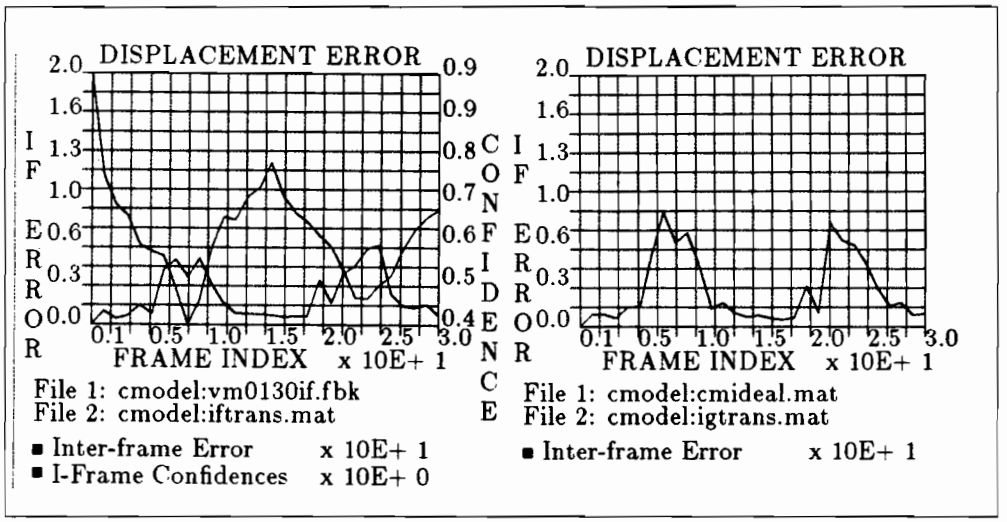

Figure 31a & 31b Range Sequence - Mean displacement error: (a) equal feature weights (b) emphasis of curvature features

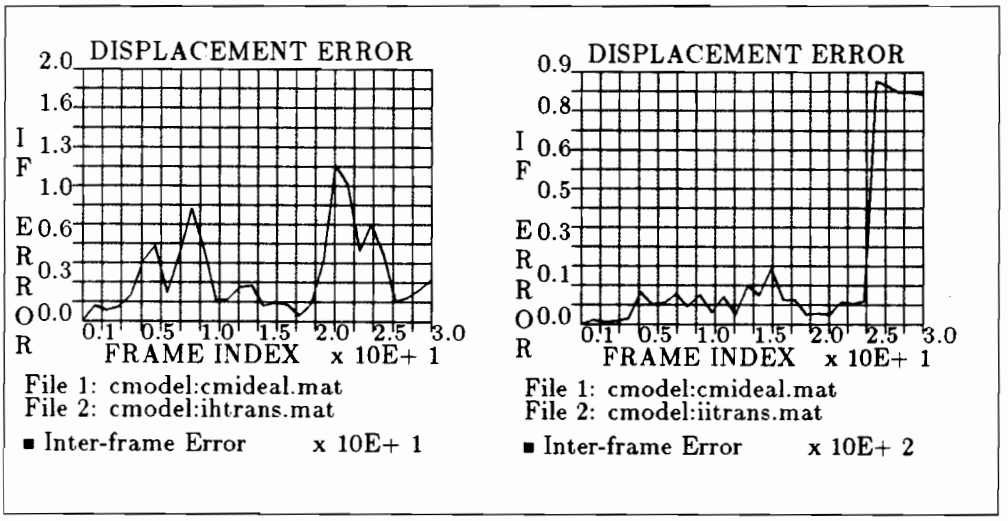

Figure 31c & 31d Range Sequence - Mean displacement error: (a) emphasis of intensity features (b) equal feature weights, small token sample size

The result shown in figure 31a is the best obtained for all experiments in the range series. The correlation functional (8) was weighted with equal emphasis on principal curvature

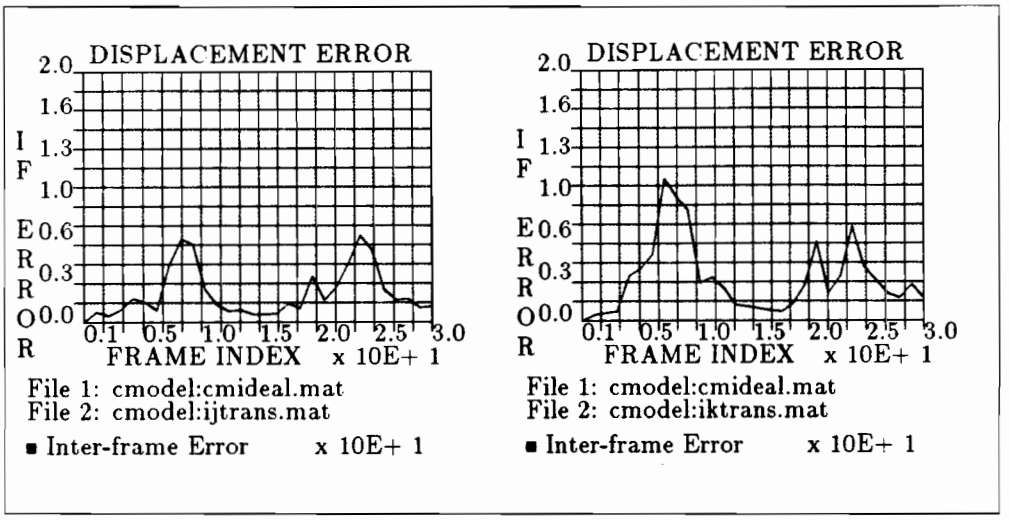

Figure 31e & 31f Range Sequence - Mean displacement error (a) equal feature weights, random selection of tokens (b) equal feature weights, small local neighbourhood size

magnitudes and the intensity gradient, and computed in an 11×11 neighbourhood. Tokens were selected on the basis of extremal features with the number of samples averaging about 300. Notice that the variation of η follows a cyclic pattern, reflecting the elliptic geometry of the model with peaks in the vicinity of the 90° and 270° views[32] which correspond to the minimal area projections. An examination of figures 30b and 30d shows that the most distinguishing features of the model from these viewpoints exist in the vicinity of occluding contours. Because the scene changes most rapidly in these regions, tokens in the vicinity of occluding contours are rejected as candidates. As a result, the algorithm defaults to a random selection of the remaining surface points where features are difficult to localize, with the peaks in η as a consequence.

Plotted against the displacement error η is the mean affinity between matching tokens as computed using (8). This parameter serves as a confidence measure for $\hat{T}_i$, and its complement an estimator for η. Notice that the mean affinity has the desirable property of tracking displacement error so as to peak at minimum error and drop to minimum value at maxi-

[32] The X ordinate of the graph corresponds to a complete 360° rotation of the model.

mum error. For example, at the 0.65 confidence level, displacement error is held to within approximately one pixel. When the confidence falls below this value, some other measure must be invoked to regain correspondence. In the example shown, this would require finding a relationship between views 5 and 12, 18 and 26, i.e. between views separated by peaks in η. This can be accomplished, for example, by computing correspondence across more global structures such as the surface patches S_i that comprise the surface S. Another approach is the so-called method of characteristic views, where an attempt is made to recognize when an object reaches a standard viewing position from which the relationship can be computed to the last known viewpoint. In either case, it is obviously important to know at which point correspondence is lost.

We now move on to the remaining experiments in this series to see how the various parameters cause η to divert from the "optimal" result. Figure 31b shows the effect of changing the weighting term w_n in (8) to emphasize curvature magnitudes only. The effect is to raise the displacement error, but only in regions that lie below confidence levels, making little overall difference. The same cannot be said, however, for a weighting that selects the intensity gradient in place of curvature magnitudes as shown in figure 31c. In this case, displacement error increases across across the entire sequence. The experiment whose error is shown in figure 31d reverts to the original feature weighting, but with the token sample size lowered to 100. As can be seen, the displacement error "blows up" (note the scale change) because least squares estimates of T_i are based on an insufficient sample population. Figure 31e confirms our hypothesis concerning token selection based on extremal features. This experiment is identical to 31a, except that token selection was based on pure random selection. The net effect is an overall increase in displacement error because $L(F_J)$ includes tokens that are more difficult to localize than those obtained through extremal feature selection. Finally, in figure 31f we modify the original experiment by changing the size of the local neighbourhood to 5×5 from 11×11. The result is a slight increase in displacement error, but not enough to make a significant difference.

4.1.2.3 Image Sequence Results

The results reported in the preceding section dealt with range data, i.e. the problem of locating surfaces was avoided by specifying their locations relative to the viewer. Consequently, errors incurred in the correspondence process can be attributed directly to the representation for surfaces and the computational model. These results might be of interest to applications involving direct surface measurement, such as the use of laser rangefinders in industrial applications. Our interest, however, is more directed towards problems in which the primary input consists of intensity images of an object in motion about the viewer. In this section we repeat the experiments whose results were shown in figures 31a-31f on a sequence of images created from the model shown in figures 30a-30d. These images were created by applying a Lambertian reflectance model to the 30 range images used in the previous set of experiments. The noise component of the resulting images varies from a minimum of 5% to a maximum of 20% with a mean value of approximately 10%[33]. Figures 32a-32f show plots of the mean displacement error obtained from the sequence of intensity images under conditions near identical[34] to those used to obtain the previous results.

The first observation to be made is that reconstruction error, i.e. the component due to the process of computing surfaces from intensity data, adds significantly to the overall displacement error. For these experiments, we used the shape-from-shading model outlined in Chapter 3, the correct application of which requires the accurate location of occluding contours. Referring again to the model shown in figures 30a-30d, notice how the surface protrusions (figures 30b & 30d) serve to confound the placement of occluding contours. In particular, figures 33a and 33b show the model and its reconstructed surface from a viewpoint that

---

[33] These values are only estimates, since we were only able to calculate exact values for simple shapes, and are most likely conservative estimates.

[34] All parameters with the exception of the A_α threshold were identical to those used in the range image experiments. The effects of varying this parameter are discussed later on in this section.

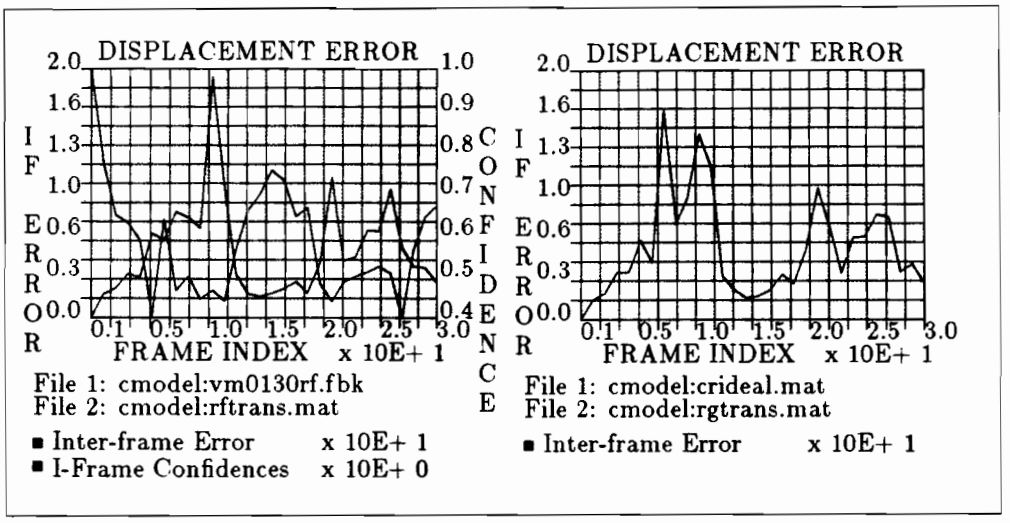

Figure 32a & 32b Image Sequence - Mean displacement error (a) equal feature weights (b) curvature features emphasized

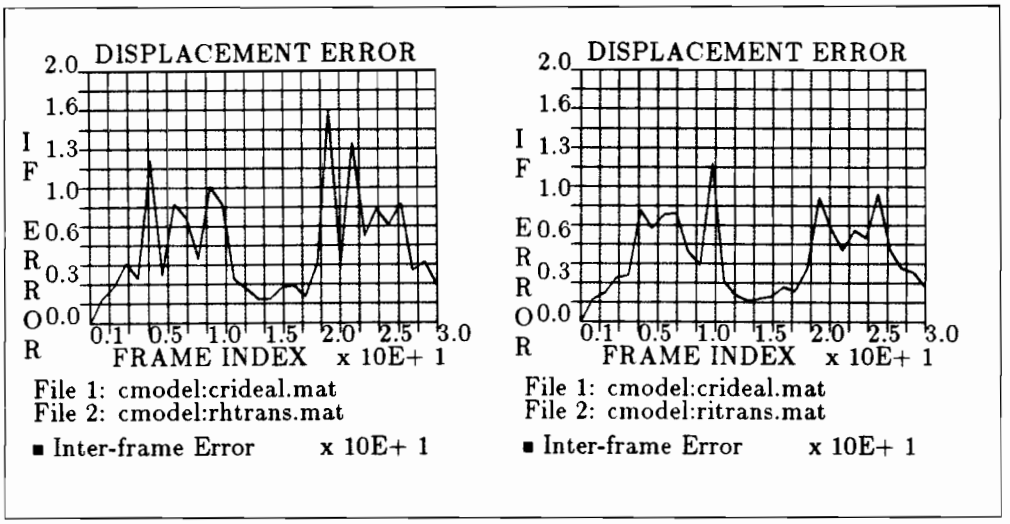

Figure 32c & 32d Image Sequence - Mean displacement error (a) intensity features emphasized (b) equal feature weights, small token sample size

highlights the problem [35]. These events often show up as additional peaks in the displacement error plot. To some extent, we gain immunity from reconstruction error by not selecting tokens that lie in the vicinity of occluding contours. However, because of the role that occluding contours play in computing the surface integral, local errors in contour placement often have

---

[35] The surface protrusions on the left half of the surface are incorrectly merged together because of the reconstruction algorithm's failure to correctly place the occluding contours.

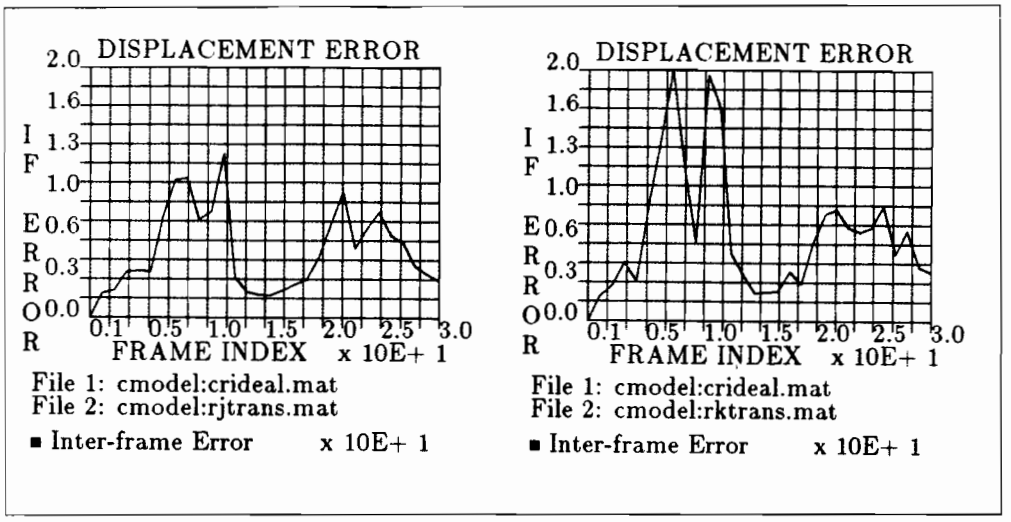

Figure 32e & 32f Image Sequence - Mean displacement error (a) equal feature weights, random selection of tokens (b) equal feature weights, small local neighbourhood size

effects that are global in nature. The errors shown in figures 33a and 33b result from not locating the depth discontinuities that separate occluding regions on the surface. This was caused, in part, by the simple line-grouping scheme employed to locate occluding contours. The limitations of such approaches and a computational model that resolves many of the difficulties associated with purely local strategies are discussed in [Zucker 1982].

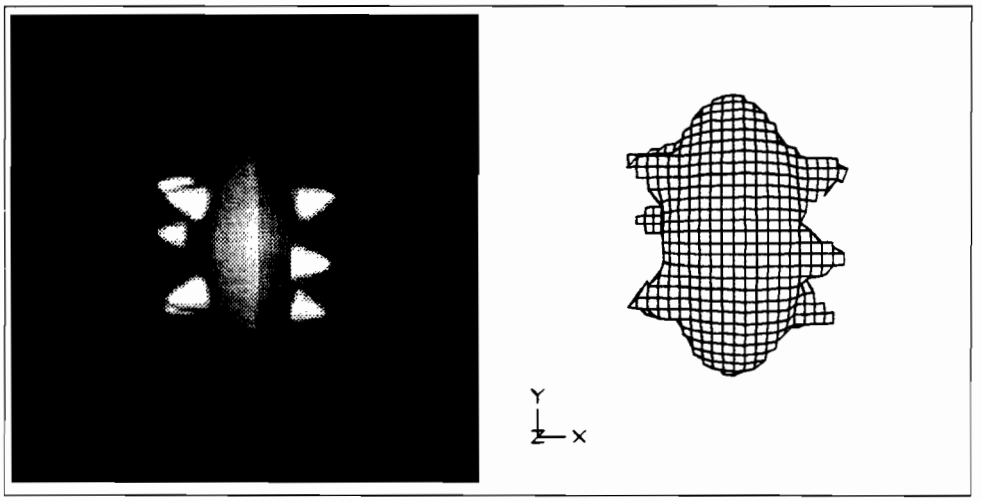

Figure 33a & 33b Reconstruction errors due to failure in localizing occluding contour

By comparing the two sets of experiments, we can observe the effects of reconstruction

error as well as the effects of parameters that govern the correspondence process. Comparison of figures 31a and 32a shows the effect of reconstruction error on the original experiment, namely in increasing displacement error in those frames in which self-occlusions are predominant. Even if the threshold for "acceptable" correspondences is raised to a mean displacement of two pixels, the range of acceptable correspondences is greatly reduced in comparison with the previous result. The effect of weighting for emphasis on principal curvature magnitudes in figure 32b has approximately the same effect as it did in figure 31b, i.e. to increase displacement error in the vicinity of peaks in η. Similarly, weighting the correlation functional (8) to emphasize the intensity gradient only (figure 32c), results in an overall increase in η across the entire sequence as it did for figure 31c. Reducing the size of the local neighbourhood and maintaining an equal weighting of features (figure 32f) has a stronger impact here than in the experiment of figure 31f. Increased noise requires stronger support from the local neighbourhood in order to maintain correspondence; reducing neighbourhood size has the anticipated effect of higher displacement error.

The most interesting results, however, are those shown in figures 32d and 32e, which correspond to a reduction in the token sample size and pure random selection of tokens respectively. Surprisingly, these yielded the best results of all experiments in the image sequence. The discrepancy is further heightened by comparing the results in figure 32d to their counterparts in figure 31d. In order to analyze events such as these, we included in the computer implementation, a facility for tracing the correspondence process in detail. The trace data provided the necessary material from which detailed analyses could be carried out.

The first observation made on comparing experiments 31d and 32d was that the A_α threshold was set lower in the image sequence experiments in order for correspondence to be more tolerant of noise in the surface reconstruction. Recall that this threshold determines the point at which correspondences are rejected on the basis of minimum affinity. A high threshold on a small sample population means that many correspondences are rejected, with the possibility of an insufficient sample on which to estimate T_i. The procedure, in order to maintain a

minimum sample size, will compensate by attempting to match randomly selected candidates from outside $L(F_j)$. Unfortunately, these tokens are usually not very highly differentiated, i.e. they do not possess extremal features that minimize uncertainty in the correspondence process. In experiment 32d, however, the lower A_α threshold resulted in fewer rejections such that the majority of correspondences were between more highly differentiated tokens.

However, this is only part of the answer because it still does not explain why the smaller sample size in experiment 32d improves the results over 32a. One would expect to do better with a larger sample size (250 vs. 100 for 32a and 32d respectively) when in fact this is not the case. An examination of trace results showed that the reason for the better performance was pure chance. The surface reconstruction in the vicinity of the 90° and 270° views is incorrect, with many prominent features corresponding to artifacts of the reconstruction. By lowering the sample size to 100, the procedure just happened to avoid tokens associated with some particularly bad surface features. The same effect was observed in experiment 32e except that the dilution was caused by randomly sampling over the entire surface.

These observations tell us that extremal features can be a double-edged sword as they not only minimize the ambiguity in correspondence but also accentuate differences caused by reconstruction error. We can, as mentioned, attempt to minimize errors by avoiding regions on the surface where they are more likely to occur, i.e. in the vicinity of occluding contours. The more important requirement, however, is to be able to detect when errors are such that the local correspondence model breaks down. For this reason is important to examine the performance estimator for T_i as well. Without such an estimator there would be no way to detect a loss of correspondence, in which case subsequent reconstruction would be totally invalid. Figure 32a includes a plot of mean displacement error vs. the confidence estimator for T_i described earlier. At a threshold of 0.61, which corresponds to a maximum displacement error η of 4 pixels, T_i estimates would be accepted for frames 1-5, 7, 13-19, and 28-30. This agrees with η as shown in all cases except for frame 8 which has a displacement error of 6.15 pixels. Fortunately, we have another constraint that we can use to detect such events,

frame-to-frame coherence [Hubschman & Zucker 1981]. This means that, on average, the visible surfaces of an object as presented to the viewer, do not change drastically during the interframe interval. A consequence of this constraint is that displacement errors, and by extension the estimator for T_i, should change smoothly. Thus, the confidence associated with frame 8 would be suspect because its left and right neighbours show a marked difference in value. In other words, $\frac{d\eta}{dt}$ must be bounded in addition to η.

We would conclude from our results that where the structure of surfaces is recovered correctly, the local correspondence model outlined is competent at recovering object motion parameters. Where this is not case, an estimate of T_i based on the correspondence of local neighbourhood structure provides a means of detecting such an occurrence. Recovery from loss of correspondence will be picked up again later in section 4.2.3.

4.1.3 Summary

The key element in the integration of descriptions from multiple views is the estimation of the set of interframe transforms T_i that map each of our viewer-centered descriptions into a common frame of reference. This in turn is dependent on a solution to the correspondence problem, which is defined as that of locating corresponding surface points across adjacent views. Our approach to the correspondence problem is to exploit the use of intrinsic feature vectors F_j in the task of selecting appropriate tokens based on extremal values of F_j. The structure of the local neighbourhood of a token, expressed in terms of the feature field defined by the set of F_j in a $u \times v$ neighbourhood of that token, provides a description that simplifies the matching problem.

Such a description is sufficiently rich in structure, that most correspondences can be solved by matching possible candidates solely on the basis of highest affinity. Because we only require a representative sample of corresponding surface points to obtain an estimate of T_i, we can afford to reject matches with low affinities and structural incompatibilities.

The result is a set of correspondences from which a reliable estimate of T_i can be made. Furthermore, the mean value of the match affinities of this set serves as a good estimator of mean displacement error η, which can be used to signal when an estimate of T_i is invalid due to a loss of correspondence.

Two sets of experiments were performed on an artificial model designed to be representative of some of the problems encountered with the correspondence, particularly that of self-occlusion. The first set of experiments used range images as input in order to isolate that component of error due to reconstruction. In this manner the correspondence algorithm and associated parameters could be examined independently of the task of localizing surfaces. The second set of experiments used intensity images as input to serve as a closer approximation to more realistic situations. In spite of errors introduced by surface reconstruction, the correspondence model was sufficiently robust as to function well in these circumstances.

4.2 Integration and Reconstruction

Having developed a solution to the correspondence problem, applicable to domains of piecewise smooth surfaces, we now consider the next step in integrating descriptions from multiple views. This is the process by which the set of interframe transforms, $T_i, i = 1, n$ is used to transform the associated set of surface descriptions, surface graphs $G_i, i = 1, n$, from a viewer-centered to common frame of reference. We call the resulting description the composite surface graph G^c. In Chapter 5 we shall see how this representation is used to infer a more abstract volumetric description of an object. For the remainder of this chapter, however, we focus on the task of computing G^c, as well as some experiments on real and artificial images. Because G^c is also an implicit reconstruction of the surfaces of an object, we can convert G^c into a polygonal model and view the results with the aid of computer graphics techniques. This provides an excellent means of obtaining feedback as to how well G^c actually reflects the object from which it was computed.

4.2.1 Computing the Composite Surface Graph

Recall that the single view surface graph G was represented by a set of intrinsic images. In the process of computing G for each view in a sequence, we end up with many sets of intrinsic images corresponding to each sample in time. We refer to intrinsic images in this context as dynamic intrinsic images (DII's). We can now outline the process of computing G^c as follows:

- For each single view surface graph G_i, compute the composite transform T_i^c that maps G_i into the coordinate frame of G^c. This is accomplished by concatenating successive transforms according to (3).
- G_i is then mapped into the coordinates of G^c by applying the composite transform T_i^c to the coordinate system of the set of intrinsic images representing G_i.
- Overlap in adjacent frames is eliminated by computing the intersection of G_i and the state of G^c before G_i is merged to G^c.

The key constraint employed in the process of eliminating overlapping surfaces when merging G_i to G^c is that *no two surfaces can occupy the same space in 3D*. The operation implied by (9) is accomplished with the aid of an auxiliary data structure called the *Object Surface Table* (or OST). It specifies the position in the common frame of each DII coordinate upon application of its composite transform T_i^c. However, the OST may also be indexed as a 3D surface, in which case the appropriate pointers provide access to features contained in DII's. As each DII is read into the OST, multiple instantiations of the same surface segment appearing in different views are eliminated in the OST, leaving a unique set of surfaces. In this manner, the composite surface graph G^c is implicitly represented as a set of DII's and an auxiliary data structure, resulting in a very compact representation for computer vision applications.

In practice, however, the composite transform T_i^c mapping a particular G_i into G^c has

associated with it an uncertainty based on the confidences attached to each T_i making up T_i^c. This means that a strict application of the "same space" constraint will not necessarily eliminate the overlap between adjacent surfaces. In recognition of this uncertainty, we extend the notion of intersection to include a small radius, δ about a point p on a surface S. In effect, the 3D space represented in the OST is quantized into voxels whose size reflects the uncertainty δ. Another problem that must be dealt with concerns the surface patches S_i[36] that comprise G^c. Recall that these surface patches are labeled as region maps in the DII's for each different viewpoint. In merging two or more descriptions of these patches, we must in effect solve a second correspondence problem, but at a more global level.

To accomplish this task, we employ a variation of the "same space" constraint which states that *if points that are sampled from two patches lying in two different frames occupy the same coordinates in 3D, then the patches must correspond.* This provides a convenient means of solving the patch correspondence problem simultaneously with determining overlap between adjacent view surfaces. As points are entered into the OST during the merging of surfaces from a new viewpoint, note is made of the patch labels to which they correspond. If a point is deemed to be non-overlapping, its corresponding patch is marked as unique, otherwise it is marked as already present. The only problem with this scheme occurs where a patch is occluded and is thus only partially visible. Using the above constraint can lead to problems where an occluded (and thus distorted) patch is accepted in place of the correct shape. We can get around this problem, though, by simply ignoring *any* patch that lies in the vicinity of an occluding contour. The assumption is that at some point the patch will become fully visible as the object moves into a more favourable viewpoint.

What we have done in the process of computing G^c is to augment each DII with three additional feature planes (figure 34), and edit the surface patch region map such that overlapping regions are deleted. Each DII is a viewer-centered representation of the visible surfaces

---

[36] The slight notational change is made to avoid confusion with the frame index i.

of an object in a particular frame. This is augmented with an additional three planes containing the corresponding coordinates in the world frame for each image coordinate. Thus, for any point in a particular DII we have immediate access to its intrinsic features, the label of the surface patch to which it belongs, and its position in the world coordinate system. The interpretation problem, however, requires that the surfaces of an object be referenced in terms of world coordinates. This requirement is met by the Object Surface Table which we used earlier as a means of enumerating surface locations. As points are entered into this structure, their position in DII space (i.e. image coordinate + frame number) is noted as well. Thus the set of DII's and the OST provide a complete representation for the surfaces of an object in a dual coordinate system.

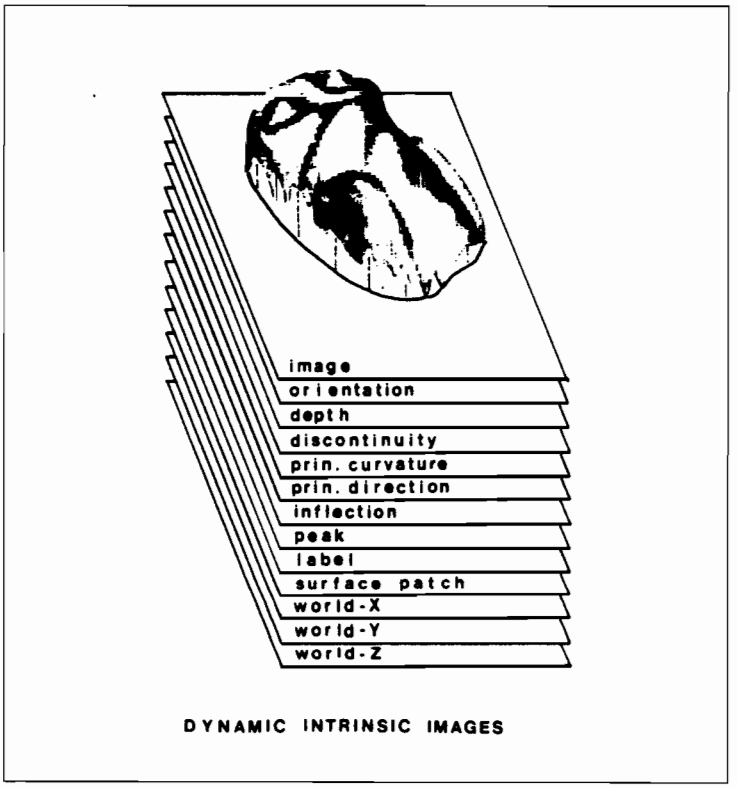

Figure 34 Additional feature planes in the set of DII's

The need for such a representation becomes clear when the problem of adjacency relationships among points in 3D is considered. In order to represent a surface in 3D with a set

of points, two pieces of information are required. These are the (x, y, z) position in space of each point, and the relationship of each point to its nearest neighbours. In our scheme, the position of each point in space is an entry into the OST, whereas the spatial relationship of a point to its neighbours is obtained by mapping from the OST into the corresponding DII where such information is readily available. A problem with the original intrinsic image concept [Barrow & Tenenbaum 1978] is the fact that interpretation is better facilitated in an object-centered representation [Marr 1982]. Note how with the addition of an additional structure, namely a transformation derived from the motion of an object, we are able to partially resolve this shortcoming. The complete solution to this problem involves the determination of the orientation of the object in the world coordinate system. This topic will be reconsidered again in Chapter 5.

4.2.2 Reconstruction of a Television Image Sequence

In order to test our representation for surfaces and the algorithms by which it is computed in a realistic environment, we performed an experiment in which a physical model was rotated and translated by hand in front of a CCD television camera. The model used was the stone owl statuette shown earlier in figure 22a, illuminated from two light sources directly above the camera lens as shown in figure 35. An 80mm telephoto lens was used such that projection of the model onto the viewing plane filled most of the available area. The owl statuette itself was painted with a flat white paint such that reflectance was approximately constant over the surface. However, markings on the surface caused by pits in the stone carving tended to alter reflectance considerably in places. A total of 31 images were created, each with a resolution of 128 $\times$ 128 pixels, and an intensity of 64 gray levels. A sample of 10 views, representing approximately 60° of rotation was then used to compute G^c. We found that because of noise and accumulated error in computing T_i^c, correspondence would break down after a sequence of 10 frames, thus limiting reconstruction to partial views of the model.

An interesting question arises as to how results can be evaluated. Lacking a 3D description

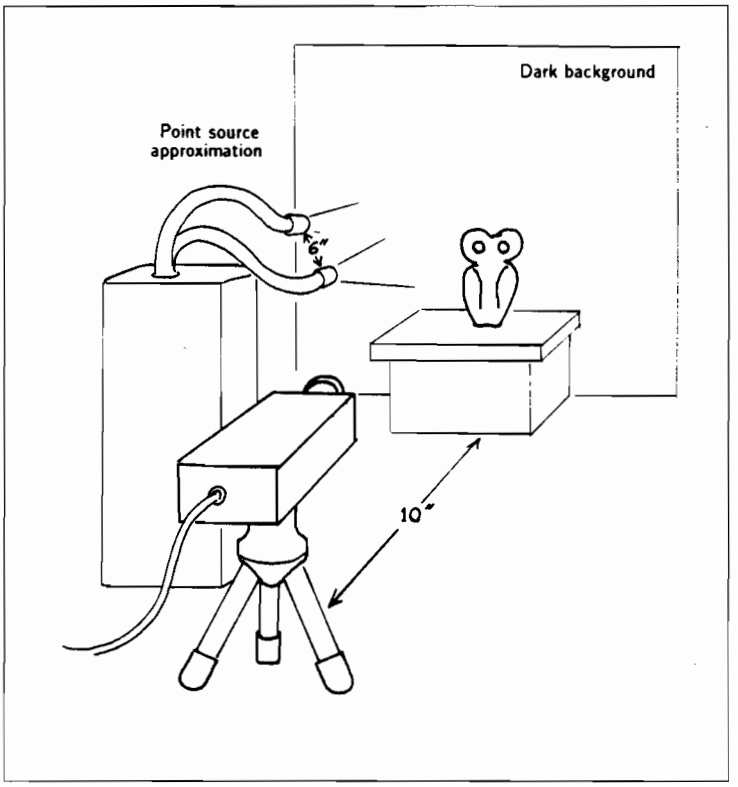

Figure 35 Imaging set-up for the owl experiments

of the model, it is difficult to come up with a simple metric for difference as we did earlier for motion parameters in the form of mean displacement error. Instead, we relied on a qualitative approach that consisted of using G^c to recreate the set of images from which the model was generated in the first place. The fidelity of G^c to the scene model is thus reflected in the difference between images used for input and those reconstructed from the model. Figures 36a-36d show the results of computing a polygon approximation to the surfaces of G^c and then displaying the result with the aid of a computer graphics display program. Due to limitations with this program, we were restricted to coarse quantizations of the surfaces represented in G^c, resulting in the smoothed rendition shown. The results, however, show that the surfaces of the owl statuette are recovered well enough for interpretation purposes.

4.2.3 Coping with Loss of Correspondence

The previous experiment underscores the need for a strategy to re-establish correspon-

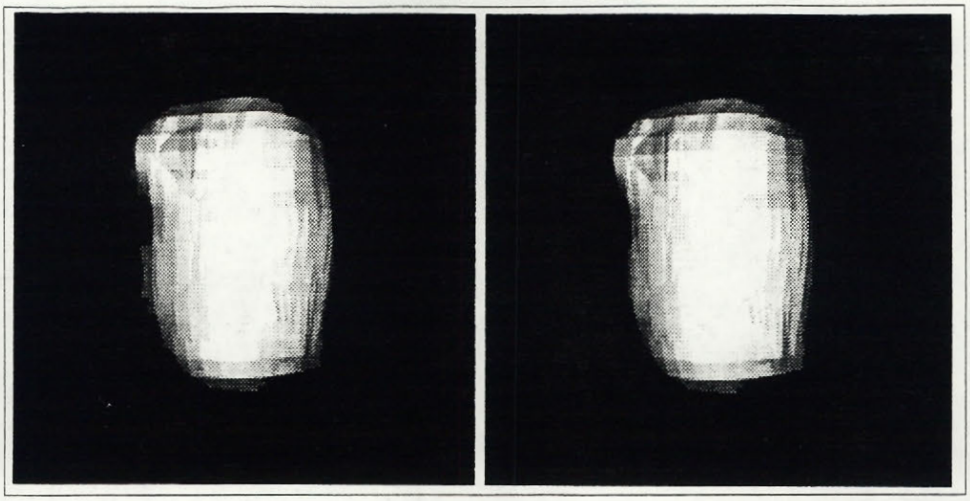

Figure 36a & 36b Reconstructed owl model - views 1 & 2

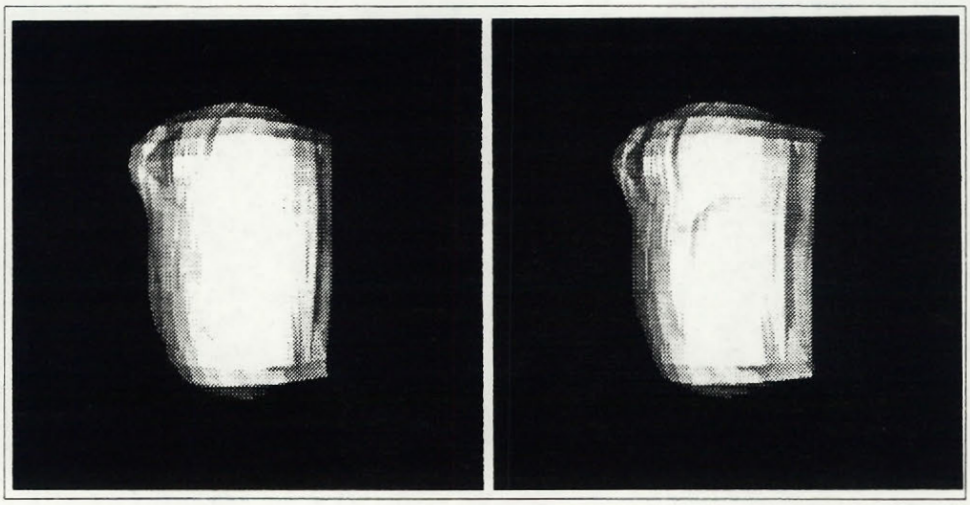

Figure 36c & 36d Reconstructed owl model - views 3 & 4

dence after it has been lost either due to a breakdown of local structure matching or an accumulation of error in the composite transform T_i^c. We refer to a sequence of views through which correspondence is maintained as a *segment*. In order to be able to successfully reconstruct the surfaces corresponding to these segments, the relationships (i.e. transformations) must be found that link the segments together. To accomplish this requires the solution of a correspondence problem, but at a more global level. At what level, then, do we attempt to solve this correspondence problem? In the context of the shape hierarchy (figure 3), we proceed from the level of feature vectors to that of surface patches, and then to aggregates

of patches in the form of sub-graphs of G^c. We could attempt an approach similar to that of Bhanu [1984] by matching on the structure of surface patches and using relationships between neighbours as constraints on the correspondence process. Such an approach would probably work well, but with a computational complexity in the present case that might not prove feasible. Consider a toy example consisting of three segments containing m, n, and o frames respectively. There are potentially $m \times n \times o + n \times o$ frame correspondences to determine in order to resolve three segments. Contrast this to the $m + n + o - 1$ correspondences we perform in order to estimate the set of interframe transforms T_i.

The above approach is equivalent to an exhaustive search over a very large space, and what is required are some constraints to help prune it. Unfortunately, such constraints are not always apparent at a local level and for this reason we seek a more global description on which to base the correspondence process. Marr [1982] pointed out that a potentially rich characterization of shape at a global level is provided by the silhouette contours of an object[37]. We can characterize a the silhouette contour of an object in frame i as a 3D space curve $\Omega_i(s)$. Our problem can now be defined as follows: given two segments $SG1$ and $SG2$, find a transformation matrix T_s mapping $\Omega_i(s)$ into $\Omega_j(s)$ for some $i \in SG1$ and $j \in SG2$. Recall that all we are looking for is the geometric relationship between *any* two frames across different segments as this is determines the set of transformations relating each element of $SG1 \bigcup SG2$.

Let us now consider how we can make use of $\Omega_i(s)$ in splicing together two segments $SG1$ and $SG2$. First note that we do not use this approach in computing the interframe transform T_i because descriptors are too global, i.e. $\Omega_i(s)$ is not stable because the silhouette contour is constantly changing as the object moves. However, given a long enough sequence, that covered by two segments for example, it might be possible to find some i and j such that $\Omega_i(s) \sim \Omega_j(s)$ for $i \in SG1$ and $j \in SG2$. This is equivalent to saying that there exists some

---

[37] The silhouette contour is defined as that which separates an object from its background.

T_s such that

$$\Omega_i(s)T_s \sim \Omega_j(s),$$

or, equivalently,

$$||\Omega_i(s)T_s - \Omega_j(s)||_{L_1} \leq \epsilon. \tag{46}$$

where ϵ specifies a threshold of similarity between the two space curves.

The computational task of finding T_s can be set up as a minimization problem in which the objective is to minimize (46). Unfortunately, the function described by (46) as one traverses the space of possible T_s's is not monotonic. Thus some effort must be made to find a set of constraints such that the search is localized to a convex region of space where a gradient search procedure can be used to find the local minimum. If these constraints are appropriately chosen, then the local minimum will, in fact, be the T_s that we are looking for. The constraints that we shall make use of are global properties of the silhouette contour by which non-suitable candidates for T_s can be eliminated before attempting correspondence. In this manner we prune the space of possible T_s's such that the local minimum in (46) is the global minimum.

Consider the images of the owl statuette shown in figures 37a-37d, with particular emphasis on the silhouette contours in each case. Notice the similarity of the contours in figures 37a to 37c and 37b to 37d, and that each pair of views is rotated by 180°. In particular, note that when similar space curves are projected onto the view plane, the areas enclosed by each will be similar as well. This turns out to be a very powerful constraint because it tells us that when projected areas are similar, the two space curves, if identical, will be related by a translation and rotation in the view plane plus a possible rotation of the view plane itself by 180°. Rather than attempt to verify the existence of T_s for each i and j in $SG1$ and $SG2$, we instead consider only those cases where the above constraint is met. These may correspond to *characteristic views* [Freeman 1978,80] of the object where the relationship between i and j is constrained to a workable subset of T_s. This is verified by searching for a rotation and translation in the view plane and possible rotation of the view plane by 180° such that T_s maps

$\Omega_i(s)$ into $\Omega_j(s)$. Of course $SG1$ and $SG2$ must be constrained to contain the appropriate characteristic views, but this is usually assured for objects that have a component of rotation, i.e. that present front and rear views.

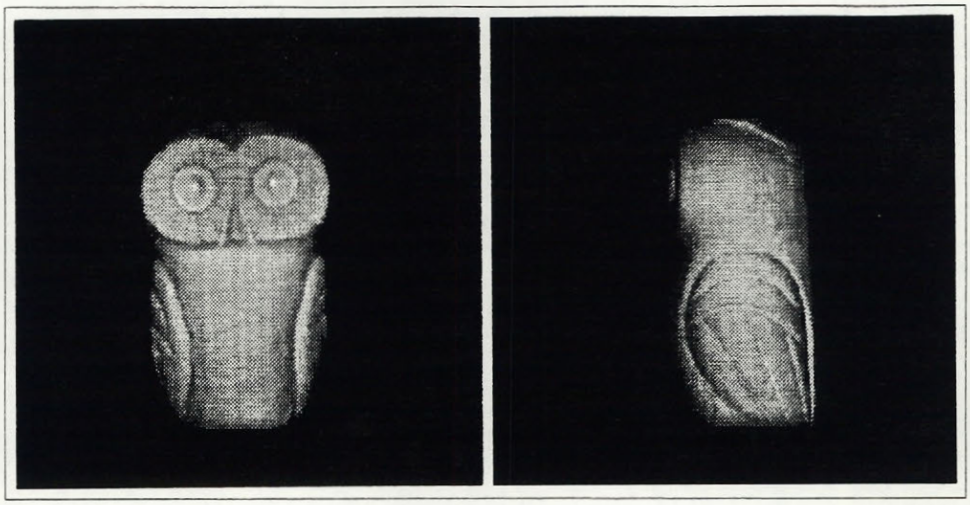

Figure 37a & 37b Owl image from the 0 & 90 degree viewpoints

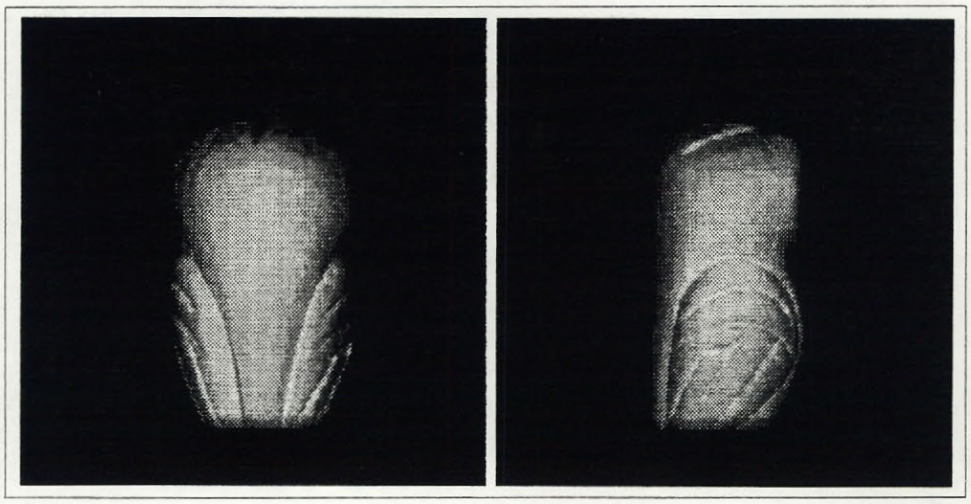

Figure 37c & 37d Owl image from the 180 & 270 degree viewpoints

The task of determining T_s is again simplified by exploiting global properties of $\Omega_i(s)$. View plane translations and rotations can be determined by finding the positions and orientations of the projections of $\Omega_i(s)$ and $\Omega_j(s)$ onto the view plane (figure 38). Positions and orientations

are defined as the centroids and major axes of the enclosed areas respectively. The procedure for determining T_s, if it exists, for a particular i and j is as follows:

1. Bring $\Omega_j(s)$ into partial correspondence with $\Omega_i(s)$ by applying a translation, $\vec{C}_i(x,y) - \vec{C}_j(x,y)$, and a rotation, $\theta_i - \theta_j$, where $\vec{C}_i(x,y)$ and $\vec{C}_j(x,y)$ are the centroids, and θ_i and θ_j are the orientations, of $\Omega_i(s)$ and $\Omega_j(s)$ respectively.
2. Translate $\Omega_j(s)$ along the Z axis such that $|\Omega_i(s) - \Omega_j(s)|$ is at a minimum[38]. If the minimum difference is less than a threshold ϵ, then T_s is deemed to exist and its parameters are determined from the view plane translation and rotation.
3. If T_s is not verified in step 2, a second iteration is performed in which $\Omega_j(s)$ is first subjected to a 180° rotation about the Y axis (i.e. rotation of the view plane). If the minimum difference is less than ϵ, the parameters of T_s are determined as before with the addition of the 180° rotation about the Y axis.

The difference between $\Omega_i(s)$ and $\Omega_j(s)$ is computed using the following functional,

$$\int_t \left(\Omega_i(s) - \Omega_j(s)\right) dt. \tag{47}$$

The area constraint together with the above procedure constitute an efficient means of determining for a particular i and j the existence of T_s, and if so, its parameters.

Figures 39a-39d show four reconstructed views of the blood platelet model shown earlier in figures 30a-30d from the identical viewpoints. Recall that we were unable to compute G^c for this object due to the breakdown of the local correspondence model at frames in which the model was heavily self-occluded. In spite of these problems, we were able to recover from the loss of local correspondence by employing the global features, procedures, and constraints described above. An examination of the results shows that the surfaces of the platelet model are recovered quite accurately. The same procedure was applied to the owl images shown in

[38] This difference will always have a minimum value when the two space curves are in closest proximity to each other.

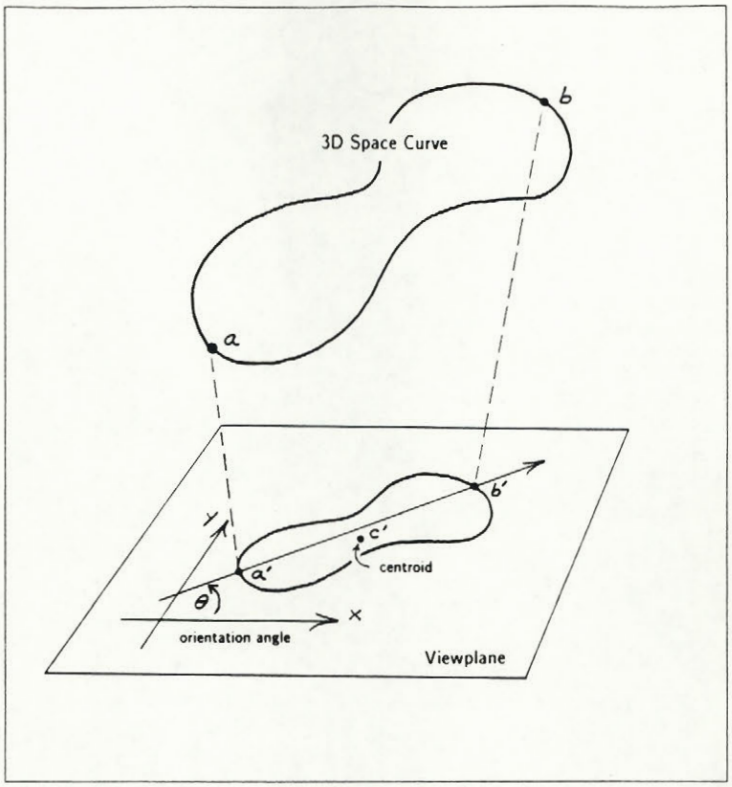

Figure 38 Orientation and centroid for a space curve projected onto a plane

figures 37a-37d, and the reconstructed surfaces are shown in figures 40a-40d, again from the same viewpoints. Unfortunately, the rendition shown is marred by smoothing and some errors in reconstructing the front surface. However, the recovered surfaces are sufficiently accurate for purposes of interpretation and coarse feature measurement. This will become evident in Chapter 5, where we use these same surface descriptions to compute abstractions of these objects in terms of ellipsoid-cylinder models.

4.2.4 Summary

In this section we discussed how the composite surface graph G^c, which is a representation for the complete surfaces of an object, is computed from a set of single view surface graphs $G_i, i = 1, n$ and a corresponding set of interframe transforms $T_i, i = 1, n$ which describe the motion of the object. This was accomplished by using the set of interframe transforms to compute a mapping from the viewer-centered coordinates of a particular G_i to a common frame

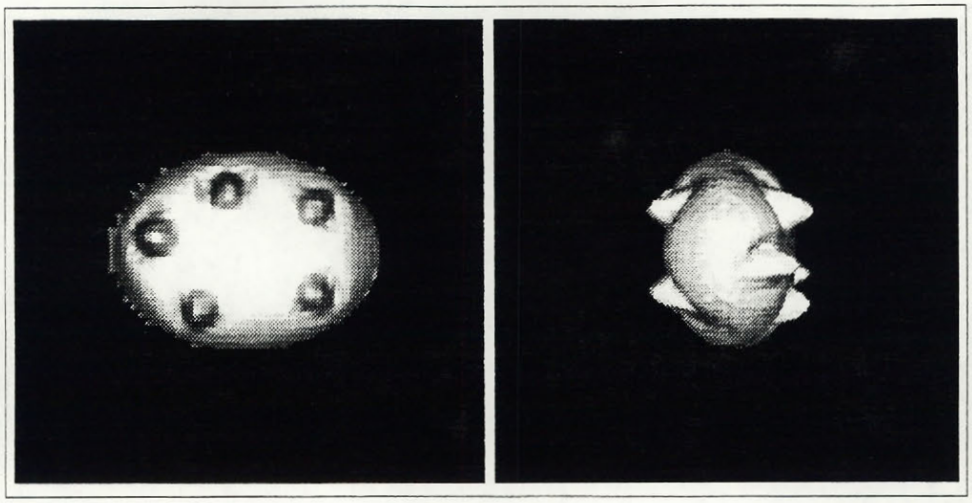

Figure 39a & 39b Reconstructed blood platelet model - 0 & 96 degree views

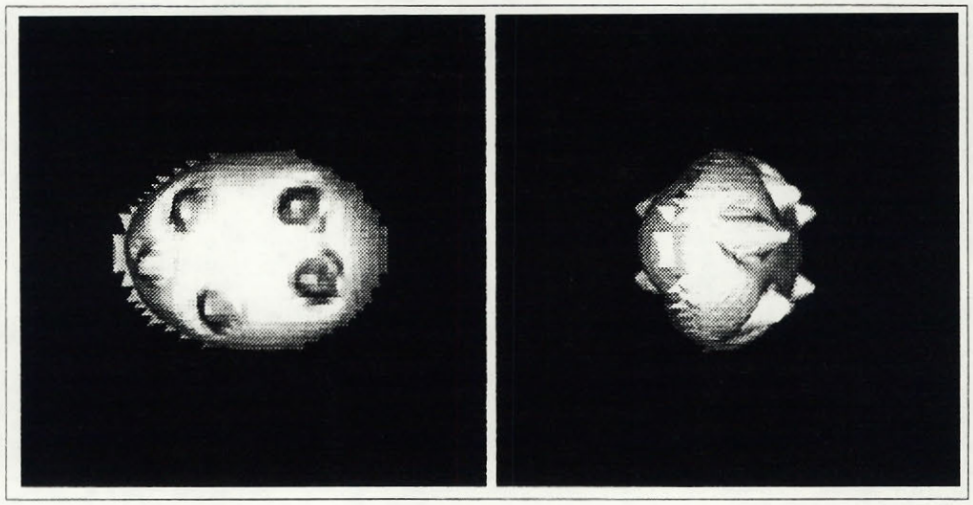

Figure 39c & 39d Reconstructed blood platelet model - 192 & 288 degree views

of reference. The physical constraint that no two surfaces could occupy the same space in 3D was used to identify overlapping views such that a unique set of surfaces could be obtained. An experiment performed on a sequence of TV images of an object in motion demonstrated the feasibility of the surface model in practical applications, provided that correspondence could be maintained. In the event that correspondence was lost, the identification of characteristic views was shown to provide a means of re-establishing correspondence. This was accomplished by exploiting constraints between matching characteristic views. Finally, the results of two experiments were presented that showed the use of characteristic views in re-establishing

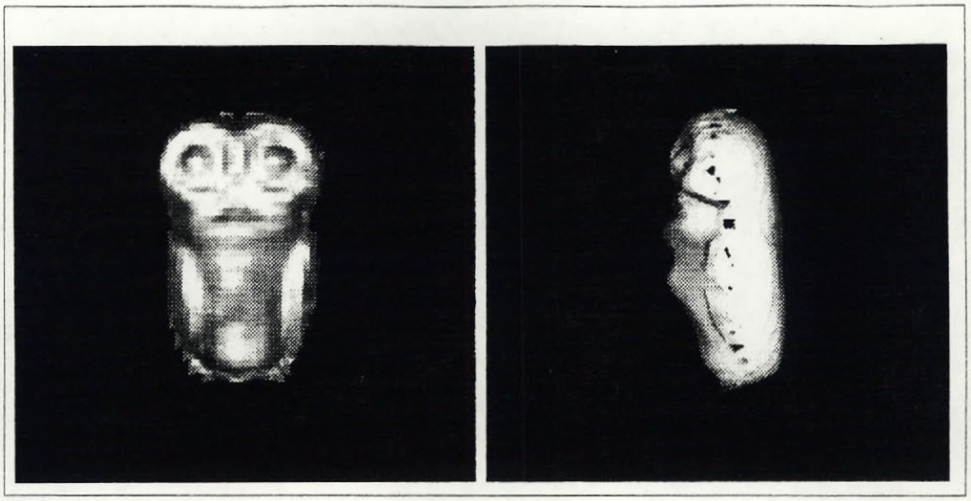

Figure 40a & 40b Reconstructed owl model - 0 & 90 degree views

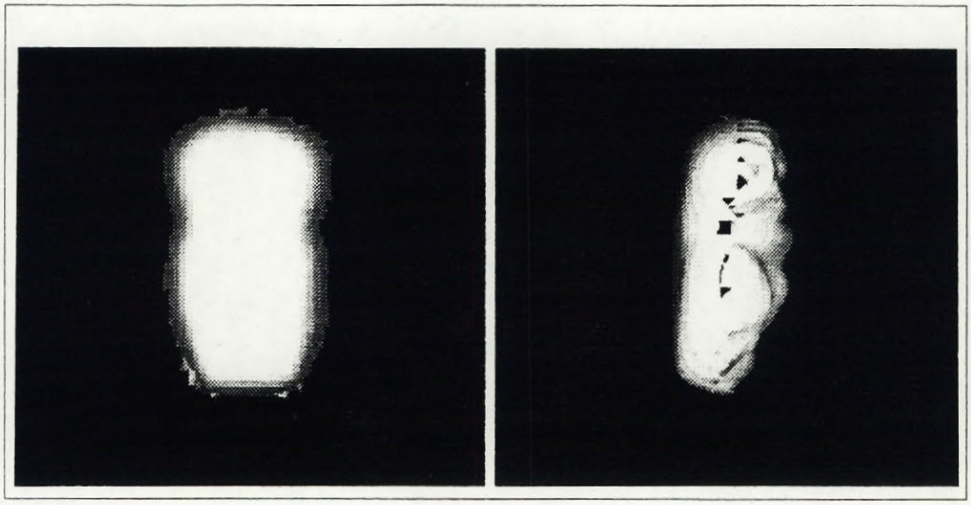

Figure 40c & 40d Reconstructed owl model - 180 & 270 degree views

correspondence.

4.3 Chapter Summary

The need to represent the complete surfaces of an object requires that viewer-centered descriptions be integrated into a common frame of reference. In order to solve this problem, we first had to determine the set of interframe transforms from which the mapping from the viewer-centered to world frames could be computed. This in turn required the solution of

the correspondence problem such that corresponding points on adjacent view surfaces could be found. Experiments presented showed that correspondence based on the structure of the local neighbourhood of a token point could be used reliably to obtain an estimate to the interframe transform matrix T_i. In addition, this structure also permitted the computation of a confidence measure for T_i, without which it would not be possible to detect the loss of correspondence. Experiments with an artificial model, designed to exhibit common problems with self occlusion, showed that loss of correspondence could be accurately detected using a confidence measure based on the correlation functional (8).

Because it is not always possible to maintain correspondence throughout a sequence of views of an object in motion, it becomes necessary to defer to an alternate strategy when this occurs. Our approach consisted of exploiting constraints on the characteristic views of an object to re-establish correspondence, making it possible to compute the composite surface graph G^c. From this representation we computed polygon models of the reconstructed surfaces which enabled their visualization as images. We used this technique to confirm qualitatively the performance of algorithms for computing both correspondence and the composite surface graph G^c. However, we still face the problem of interpreting the surfaces of G^c as the object that they represent. In the following chapter, we shall see how the description contained in G^c provides a basis for solving the problem of geometric inference.

Chapter 5 From Surfaces to Objects

The final component of the shape problem as defined in Chapter 2 is that of inferring the geometric structure of an object from its surfaces. But what exactly do we mean by the term "geometric structure"? The answer to this question is largely dependent on how objects are represented, for example in a CAD/CAM database that contains a description of an object for manufacturing. Such representations vary widely in scope [Badler & Bajcsy 1978, Requicha 1980] depending on what needs to be represented in a given application. In manufacturing, representations are often tied to physical processes, such as the description of an object in terms of the volume swept by a numerically controlled milling machine in fabricating components of the object. A similar line of reasoning can also be taken with natural objects, i.e. that processes responsible for growth and formation give rise to representations from which such objects can be identified [Witkin & Tenenbaum 1984]. An early example is found in the Generalized Cylinder concept, where limbed objects such as found in nature are represented by conjunctions of generalized cylinders[39] [Binford 1971, Marr 1982]. In human and animal figures, limbs serve to mark a natural decomposition of the form into generalized cylinders of appropriate size [Marr 1982].

Thus, in order to discuss the problem of geometric inference, we must do so in the context

---

[39] A generalized cylinder differs from the usual definition by allowing the radius to vary as a function of position along the central axis. For example, a piece of wood turned on a lathe is an example of a generalized cylinder.

of a particular representation. In this chapter we consider the problem in terms of generalized cylinder models because of their historical significance. However, the application of our work is also directed towards problems of shape analysis in cell biology where generalized cylinder models do not suffice as a representation for cell morphology. For this reason we also focus on an ellipsoid-cylinder model where cells, blood platelets in particular, are represented as oblate spheroids with cylindrical projections [Frojmovic & Milton 1982]. While these two domains differ significantly, many of the same problems are common to both. We see these problems as the following:

- Primitive Instantiation: the task of locating model primitives such as generalized cylinders within an image or scene.
- Primitive Conjunction: the task of parameterizing each instantiated primitive and piecing together these primitives into a global representation for an object.
- Model Interpretation: the task of interpreting the conjuncted model as some object in a particular domain.

In the first part of this chapter, we look at the above problems in the context of the composite surface graph G^c and show how a model composed of ellipsoids and cylinders is extracted from this representation. The second part of the chapter presents the results of some experiments in computing ellipsoid-cylinder models from three sets of data: the artificial blood platelet model, microscopy images of a blood platelet in motion, and TV images of the owl statuette. Although the geometric inference paradigm is presently limited to a simple vocabulary of shapes (i.e. ellipsoids and cylinders), the renditions obtained in our experiments suggest that the approach can be generalized by extending the set of primitive model elements.

5.1 Interpreting the Composite Surface Graph G^c

The motivation behind the surface graph concept was to make explicit the structure of surfaces in such a way so as to facilitate the problems of correspondence and geometric

inference. This led to a notion of "curved edges", or extrema in surface normal curvature by which the surfaces of an object were decomposed into a set of homogeneous regions or patches. Now if the object that is composed of these patches is assumed to admit a decomposition into volumetric primitives, it would be reasonable to assume further that each patch lies on the surface of only one of these primitives. The only way in which a patch could span the surfaces of two or more primitives would be for it to contain the discontinuities, peaks, or points of inflection marking the intersections between different components, in violation of the homogeneous definition of a patch. Note, however, that several patches may span a single primitive as in the case of a cylinder which is composed of three patches according to our definition. Because we have a reasonable assurance that there is at least a one-to-one mapping between the patches in G^c and the volumetric primitives comprising the object, a sound basis exists from which to infer the local geometric structure of the object from G^c.

5.1.1 Primitive Instantiation

The first stage of interpreting G^c involves the identification of the primitives associated with each S_i of G^c. One method, reported by Hall et al. [1982], consists of fitting the surface to be identified with a general quadratic equation, and making an interpretation on the basis of invariant properties of the resulting parameters. For the following quadratic,

$$a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{13}xy + 2a_{23}zy + 2a_{14}x + 2a_{24}y + 2a_{34}z + a_{44} = 0, \quad (48)$$

the authors computed the seven quantities listed as follows:

$$I = a_{11} + a_{22} + a_{33} \quad (49)$$

$$J = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} + \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} + \begin{vmatrix} a_{33} & a_{31} \\ a_{13} & a_{11} \end{vmatrix} \quad (50)$$

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \quad (51)$$

$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} = A_{44} \tag{52}$$

$$A' = A_{11} + A_{22} + A_{33} + A_{44} \tag{53}$$

$$A'' = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{vmatrix} + \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} + \begin{vmatrix} a_{22} & a_{24} \\ a_{42} & a_{44} \end{vmatrix} + \begin{vmatrix} a_{33} & a_{34} \\ a_{43} & a_{44} \end{vmatrix} \tag{54}$$

$$A''' = a_{11} + a_{22} + a_{33} + a_{44} \tag{55}$$

where A_{ij} are the cofactors of (52).

The geometry of the surface is then classified on the basis of these quantities according to the table in Appendix D. While Hall et al. applied this technique to the classification of global surface shape, its local application will suffice for the purpose of instantiating the geometric primitive associated with a particular surface patch. In addition, the authors were able to determine the orientation of the object in space by obtaining the rotation and translation parameters of the quadratic by eigenvalue analysis, in a manner similar to that presented in Chapter 2.

In the case of the composite surface graph G^c, information about local surface characteristics is already made explicit in terms of principal curvatures. We can easily determine the primitive associated with a particular surface patch by noting the signs of the principal curvatures at selected points on the surface. A patch lying on a cylindrical surface, for example, is identified by the presence of zero curvature along one of the principal directions. Similarly, patches corresponding to the surfaces of an ellipsoid will have the ratios of their principal curvatures reflect the eccentricity of the ellipsoid. A hyperboloid will exhibit similar curvature characteristics as the ellipsoid with the exception of opposite signs in the principal curvatures. This classification is summarized in figure 41a.

In practice, however, measurements of curvature are often subject to noise and quantization error. For example, small variations on a planar surface are sufficient to introduce artifactual

positive and negative curvatures leading to an incorrect classification of the surface. One approach to this problem would be to apply a threshold to the principal curvature magnitudes. The absolute values of curvature magnitudes would have to exceed this threshold in order to be labeled as positive or negative respectively. In fact, we used this approach earlier to detect extremal values in surface normal curvature at the surface decomposition stage. As a consequence, however, the surface patches S_i are not truly homogeneous. For this reason, we need to invoke statistical considerations in making inferences about the global characteristics of a patch from local surface measurements.

We found that a useful statistic in this regard is the mean value of the ratio of the principal curvatures computed at each point of S_i. This is used according to figure 41a as the primary classifier for the primitive type associated with a particular S_i. Figure 41b shows a surface composed of an ellipsoid and five cones. Because the cones are ruled surfaces, one would expect the mean curvature ratio to be high for patches corresponding to the cones, and low for the remaining patch corresponding to the ellipsoid. Curvature statistics computed from this surface are shown in figure 41c, and confirm our expectations. We can use this procedure to make reliable estimates of surface type, even in the presence of noise, from data already made explicit in G^c.

Model	Signs of Prin. Curv.	Ratios of Prin. Curv.
Sphere	(+,+),(-,-)	=
Ellipsoid	(+,+),(-,-)	$\propto$ eccentricity
Cylinder	(+,0),(-,0)	∞
Hyperboloid	(+,-)	$\propto$ eccentricity
Parabolic Planar	(0,0)	=

Primitive Model Association

Figure 41a (a) Surface patch classification based on ratios of principal curvatures

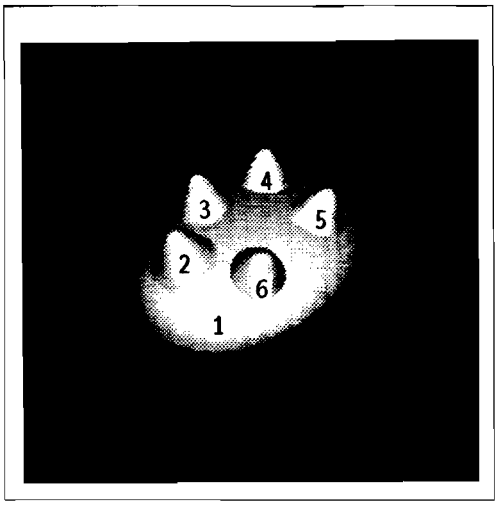

Figure 41b Surface composed of an ellipsoid and 5 cones

Patch	Mean Curvature Ratio	Histogram of Prin. Curv.
1	0.2956	
2	5.0626	***
3	5.5218	***
4	6.8480	****
5	7.1003	****
6	73.9856	*******∫∫*******

Discrimination: Mean Curvature Ratios

Figure 41c Surface patch classification of figure 41b

5.1.2 Primitive Conjunction

The second stage of geometric inference consists of computing the parameters for each primitive instantiation of the first stage and then taking the conjunction of all primitives. Included in the process is the task of resolving multiple instantiations of the same primitive. This is caused by having several surface patches map to the same primitive as in the case of a cylinder, or because of error in reconstructing the surfaces of an object. For example, additional patches are created in the event of "holes" on the reconstructed surface, resulting from self occlusions that go unresolved. We begin by looking at the parameterization problem, and then

consider how the resulting parametric models can be used to resolve multiple instantiations. This leads to a composite model of an object in terms of the remaining volumetric primitives.

5.1.2.1 Parameterization

The task of parameterization is straight-forward once each patch S_i of G^c has been labeled as an ellipsoid or cylinder. This is accomplished by fitting the appropriate parametric model to the surface represented by each patch. One method would be to use the approach of Hall et al. [1982], described in the previous section, concurrently while determining patch labelings. Our approach is similar, but more direct in that the type of model being fit is determined a priori from curvature information. We proceed by first determining the orientation of the surface in space by computing the intrinsic coordinate system as discussed in Chapter 2. The surface is then rotated and translated into standard position from which a least squares approach can be used in fitting the appropriate model.

The second order polynomial,

$$z^2 = \alpha x^2 + \beta y^2 + \gamma, \tag{56}$$

is sufficient for determining the parameters of an ellipsoid in standard viewing position, (i.e. axes parallel to the coordinate axes). The parameters of an ellipsoidal primitive,

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1, \tag{57}$$

are expressed in terms of (56) as follows:

$$a = \sqrt{\frac{\gamma}{\alpha}} \qquad b = \sqrt{\frac{\gamma}{\beta}} \qquad c = \sqrt{\gamma}. \tag{58}$$

A cylindrical primitive is treated in much the same way, i.e. an estimate of the axis of symmetry is computed from the eigenvectors of the inertia tensor that is estimated from points on the surface. This approach will not work, however, if there is insufficient symmetry

in the surface sample. Cylinders are often difficult to deal with because their surfaces tend to be obscurred as the axis of symmetry approaches the orientation of the normal to the viewing plane[40]. Keeping this problem in mind, we first attempt to parameterize the cylinder by computing its ICS and mapping surface points to a standard viewing position. The cylinder is then approximated as a parallelepiped volume from which the two vertices and radius can be determined. Partial views are accomodated by treating the cross-sectional dimensions of the parallelepiped as equal, i.e. the cylinder radius is taken to be the larger of the two dimensions. The axis of symmetry is identified as that corresponding to the minimum moment of inertia in the ICS, and the cylinder endpoints are easily located by determining the intersection of this axis and the parallelepiped approximation.

Next, we must determine if the above parameterization is correct, in particular if the axis computed is really the axis of symmetry (it will not be if the surface sample lacks the required symmetry). In situations such as these it is often useful to default to local surface analysis. Because a cylinder is a ruled surface [Hilbert 1952], the directions of minimum principal curvature will be parallel to the axis of symmetry. If these directions are not compatible with the computed axis, we know that the parameterization is invalid. An attempt is then made to estimate the position and radius of the cylinder from the curvature of the surface in the direction of maximum principal curvature. This approach is similar in concept to the one used by Nevatia & Binford [1977], but we did not include it in our implementation. As a result, we had difficulty in estimating the orientation of small cylindrical surface patches (this will be illustrated later on in the experiment with the owl data).

Each ellipsoid resulting from the parameterization process will be represented by 9 parameters, with 3 corresponding to the centroid $< X_c, Y_c, Z_c >$, 3 to the direction cosines of the ICS, and the remaining 3 to the sub-axis lengths computed from the polynomial approximation

---

[40] Ellipsoids, on the other hand, are more forgiving since their shape is relatively unambiguous for all points of view.

of the surface patch (56). A cylinder is represented by 7 parameters, with 6 corresponding to the two endpoints $< X_1, Y_1, Z_1 >$ and $< X_2, Y_2, Z_2 >$ along the principal axis of the ICS, and the remaining parameter to the radius R. In computing these parameters, the implicit assumption is made that points at which the surface function is sampled are distributed evenly over the entire surface, i.e. that estimates are based on complete surfaces. This is rarely the case, however, as is illustrated by the example in figure 42, which shows the surfaces formed by the intersection of an ellipsoid and a sphere. Each of the three patches in this example is incomplete because its surface is partially occluded by the intersection. In the ideal case, where the locations of points on a surface are known exactly, one would expect to be able to accurately determine the parameters of a geometric model from a partial view of its surfaces. This is especially true if the surface in question is symmetric about the visible and occluded portions. The situation changes, however, when there is uncertainty in the position of points on a surface due to the presence of noise. In such cases, parameter estimates may vary considerably from those obtained using the complete surfaces of an object.

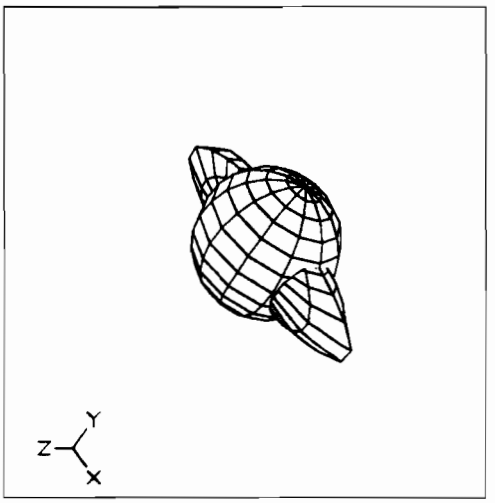

Figure 42 Surface formed by the intersection of an ellipsoid and a sphere

To illustrate this effect, we repeated an earlier experiment (whose results were summarized in Table 2) in which the parameters of an ellipsoid were recovered from shaded images. Instead of using two views showing all visible surfaces, we repeat the experiment using only single

view in which half the surface area is occluded. The results are summarized in Table 5 (Appendix C). A comparison of the parameters obtained from complete and partial surfaces shows that in both cases, position and orientation are recovered quite well. However, in the case of the axis length parameters, i.e. those obtained via a polynomial fit to the surface after transformation to standard position, the results are disappointing. For this reason, we first attempt to minimize error by using as much information as is available from multiple views (i.e. from all visible surfaces). Where this is not possible, an attempt is made to impose symmetry considerations (such as was done in the case of the cylinder using the parallelepiped approximation), but it is not always clear how such constraints should be applied.

The practical implication for the geometric inference paradigm is that parameter recovery error will be proportional to the surface area of the geometric primitive *not* spanned by a patch S_i. In addition, any unresolved self-occlusions also contribute to recovery error. By increasing the amount of information available to the process through multiple views of an object, we attempt to increase the certainty of interpretation. While partial view interpretation is often sufficient to distinguish one object from another in a particular domain, other applications require accurate recovery of object geometry. The blood platelet problem is a case in point because, in addition to the classification of cells, measurements of parameters based on cell geometry are also of interest. The same is also true for robotics applications in which vision is used as feedback to a manipulator system. An accurate rendering of size and shape simplifies the path planning problem by reducing the uncertainty of object positions in the workspace [Gupta 1986].

5.1.2.2 Conjunction

The next step after locating and parameterizing model primitives is to organize this data into a structure that facilitates interpretation. However, before doing so, multiple instantiations of the same primitive have to be removed from consideration. As mentioned earlier, these arise from errors in computing G^c where a patch may be split in two due to an unresolved

occlusion. One method of detecting multiple versions of the same primitive is to check for intersections between newly discovered primitives and those found up to that point. Again, as in the case of surfaces, no two identical primitives may occupy the same space in 3D. The problem, however, is in discriminating valid intersections between adjacent primitives and those due to multiple instantiations. An extreme case is that where an object is composed of identical primitives (figure 43). The pair of cylinders on the left are deemed to be different cylinders because of the difference in the relative orientations of their principal axes. The pair on the right, however, are taken to be multiple instantiations of the same cylinder primarily because of the alignment of their major axes, and also because of their identical radii.

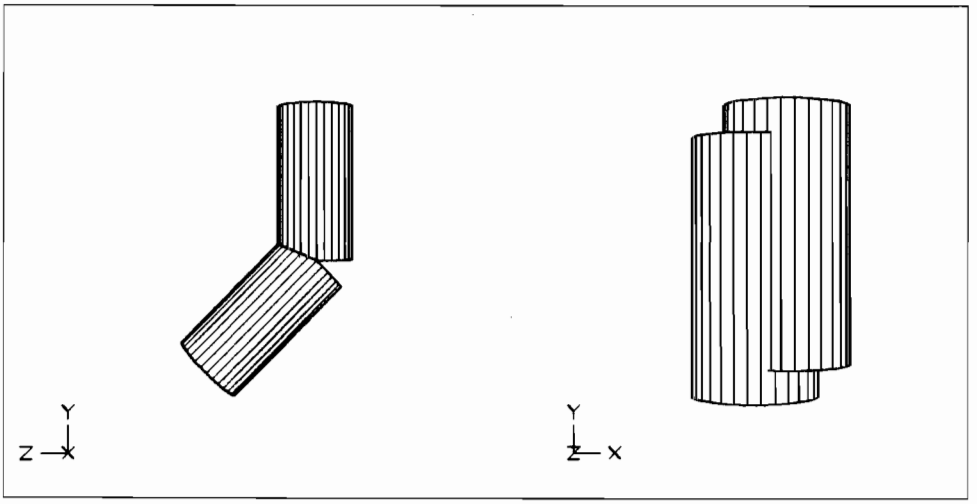

Figure 43a & 43b Valid (a) vs invalid (b) intersections

One solution to this problem would be to compute the common volume occupied by two primitives of the same type. This would serve as a measure of similarity from which multiple instantiations could be distinguished, but at a cost of computational complexity. A simpler means of achieving the same end would be to use a measure of similarity involving the parameters describing each primitive. We defined two similarity functions, $D_e(V_i, V_i')$ for

an ellipsoid,

$$\begin{aligned} D_e(V_i, V_i') = & \alpha_e \sqrt{(X_c - X_c')^2 + (Y_c - Y_c')^2 + (Z_c - Z_c')^2} + \\ & \beta_e \left(|(\theta_x - \theta_x')| + |(\theta_y - \theta_y')| + |(\theta_z - \theta_z')| \right) + \\ & \gamma_e \left(|(a - a')| + |(b - b')| + |(c - c')| \right) \end{aligned} \quad , \tag{59}$$

where $< X_c, Y_c, Z_c >$ and $< X_c', Y_c', Z_c' >$ are the centroids, $(\theta_x, \theta_y, \theta_z)$ and $(\theta_x', \theta_y', \theta_z')$ are the direction cosines of the ICS, and (a, b, c) and (a', b', c') are the axis lengths of elliptical volumes V_i and V_i' respectively,

and $D_c(V_i, V_i')$ for a cylinder,

$$\begin{aligned} D_c(V_i, V_i') = & \alpha_c \sqrt{(X_1 - X_1')^2 + (Y_1 - Y_1')^2 + (Z_1 - Z_1')^2} + \\ & \beta_c \sqrt{(X_2 - X_2')^2 + (Y_2 - Y_2')^2 + (Z_2 - Z_2')^2} + \\ & \gamma_c \left(|(R - R')| \right) \end{aligned} \quad , \tag{60}$$

where $(< X_1, Y_1, Z_1 >, < X_2, Y_2, Z_2 >)$ and $(< X_1', Y_1', Z_1' >, < X_2', Y_2', Z_2' >)$ are the endpoints and R and R' are the radii of cylindrical volumes V_i and V_i' respectively.

The weighting parameters α, β, and γ reflect the certainty to which the position, orientation, and sizes of the primitives being compared are known, and are determined experimentally. In practice there is usually enough variation in the geometry of an object such that multiple instantiations of the same primitive stand out[41] unambiguously in terms of these similarity measures.

Once a unique set of geometric primitives has been inferred from the surface patches of G^c, this data must be organized into a global representation in order to facilitate interpretation and measurement. We adopted the generalized cylinder approach of Marr [1982] and his colleagues in which data is organized at a hierarchy of scales. The principal advantage of this

---

[41] Two instantiations are deemed to be identical if the similarity function returns a value less than a specified threshold, which is determined experimentally.

representation comes under the heading of what Marr called "stability and sensitivity". This means that the representation should reflect the similarity between two like shapes while also preserving the differences between them. In our case, this hierarchy is ordered on the basis of size and organized in a tree structure as shown in figure 44. The procedure for computing this representation is quite straightforward and will not be dealt with in detail.

Basically, the algorithm seeks to find the largest geometric primitive which then serves as the trunk of the resulting tree structure. The tree is filled out by recursively scanning the list of instantiated primitives for elements adjacent to the root node. Each node contains the list of parameters describing the associated primitive. Links between nodes describe the relative orientations between adjacent axes of symmetry (figure 44). The procedure iterates until the list of primitives is exhausted, at which time the tree is completely filled out. From a computational viewpoint, the most difficult task in the above process is determining which elements in the list of primitives are adjacent. This involves determining whether or not two primitives intersect and requires a maximum of n^2 tests where n is the number of primitives in the list. Since the number of primitives was small in our case, we verified each primitive exhaustively and did not attempt to optimize the procedure in any way. For example, the number of comparisons can be reduced by restricting the search spatially. The immediate vicinity of a primitive can be represented by a spherical approximation determined from its parameters. Primitives with non-intersecting envelopes can be eliminated rapidly on the basis of a simple distance measurement.

5.1.3 Interpretation

The final component of the geometric inference problem is that of interpreting the geometry of an object as reflected in the now organized set of geometric primitives which we call an ellipsoid-cylinder model. We perform this task in one of two ways:

- By comparing the ellipsoid-cylinder model to a set of pre-stored models describing various

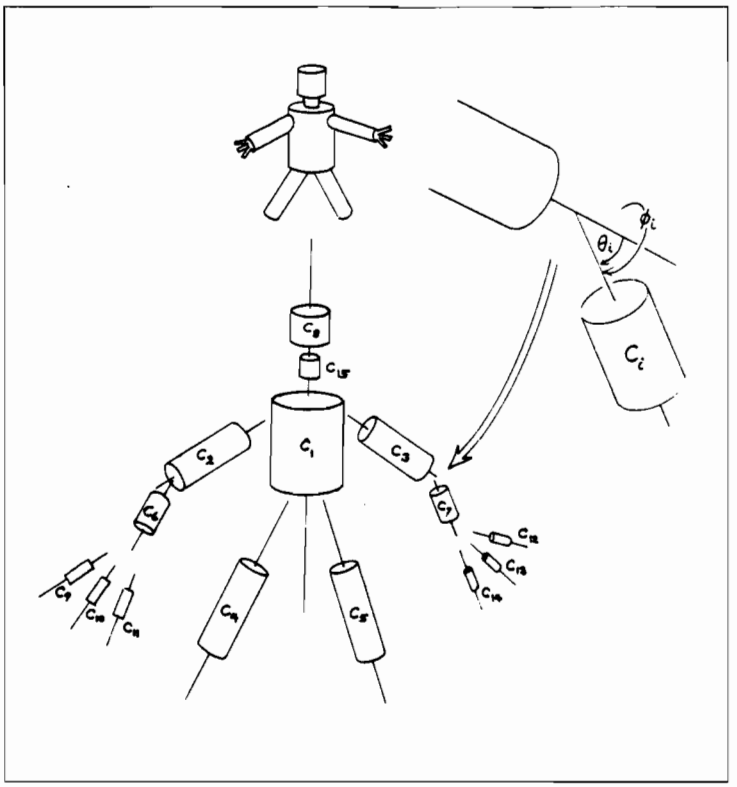

Figure 44 Hierarchical organization of geometric primitives

objects that exist in a domain.

- By extracting feature descriptions from the ellipsoid-cylinder model such that general properties can be compared in a more abstract form.

Comparison of a computed to a stored model is facilitated by virtue of a hierarchical organization and proceeding from coarse to fine scales, where a failure to match at a coarse scale precludes further attempts to do so at finer scales. One possibility would be to make the comparison on a primitive by primitive basis with a measure of similarity that reflects differences between the structure of two primitives as well as between that of their immediate neighbourhoods[42]. In order to compare a model to a prototype, however, a global figure of merit is required that reflects the similarity between two models. This is necessary because

[42] This assumes, of course, that correspondences between components of the computed model and its stored prototype can be determined, and is discussed further on.

exact matches between an unknown model and a prototype are unlikely, due of the generic nature of a prototype, i.e. it reflects the general characteristics of a class and not specific instances of particular objects.

Another motivation for a global measure of similarity involves cases in which *only* partial views of the surfaces of an object are available. Enough information may be present in an ellipsoid-cylinder model computed from partial views to make a distinction between the different class prototypes. This would also provide a means of dealing with "noisy" models where the computed geometry is distored by errors in the reconstruction process. A figure of merit reflecting global similarity is the following weighted sum,

$$M = \sum_{i=1}^{n} H(i) D^i(V_i, V_i'), \qquad (61),$$

where $H(i)$ is a weighting term that is inversely proportional to V_i's level in the hierarchy[43], and $D^i(V_i, V_i')$ is a similarity function between a computed primitive V_i and its corresponding primitive V_i'. The function returns a value between 0 and 1 reflecting the similarity between the two primitives (in this case 1 denotes exact similarity).

$D(\cdot)$ differs from $D_e(\cdot)$ and $D_c(\cdot)$ that were used earlier in eliminating multiple instantiations, and is composed of two parts. One reflects a comparison between the properties V_i and V_i' (similar in function to $D_e(\cdot)$ or $D_c(\cdot)$), and the other a comparison between properties of the neighbourhoods of V_i and V_i'. Because the computed model is arbitrarily scaled in size and at no specific position and orientation in space, comparisons must be based on invariant properties. For this reason we cannot apply D_e and D_c. Instead, comparisons are based on the relative dimensions of each primitive, i.e. the ratios of the principal axis lengths for an ellipsoid and the ratio of length to radius for a cylinder. The description of the neighbourhood of V_i is given in terms of what Marr [1982] called "adjunct relations", which define the relative orientation between the major axes of two adjacent primitives. In addition, this relation may

[43] i.e. to the size or volume of the primitive element.

also be expanded to include relative dimensions in terms of axis ratios. A measure of similarity between the neighbourhoods of two primitives, V_i and V_i', can be expressed as a weighted sum of differences for each adjunct relation. Combining the two similarity components yields a metric that is invariant to both size scaling and arbitrary position and orientation in space. For an ellipsoid,

$$\begin{aligned} D^i(V_i, V_i') \quad = \quad & \alpha \left(\frac{\frac{a}{b}\frac{a'}{b'} + \frac{b}{c}\frac{b'}{c'} + \frac{a}{c}\frac{a'}{c'}}{\sqrt{\frac{a^2}{b^2} + \frac{b^2}{c^2} + \frac{a^2}{c^2}}\sqrt{\frac{a'^2}{b'^2} + \frac{b'^2}{c'^2} + \frac{a'^2}{c'^2}}} \right) + \\ & \beta \left(\frac{\sum_j \theta_{i,j}\theta'_{i,j}}{\sqrt{\sum_j \theta_{j,i}\theta_{j,i}}\sqrt{\sum_j \theta'_{j,i}\theta'_{j,i}}} \right) + \\ & \gamma \left(\frac{\sum_j \phi_{i,j}\phi'_{i,j}}{\sqrt{\sum_j \phi_{j,i}\phi_{j,i}}\sqrt{\sum_j \phi'_{j,i}\phi'_{j,i}}} \right) \end{aligned} \quad , \tag{62}$$

where (a, b, c) and (a', b', c') are the major axes, and $\theta_{j,i}$, $\theta'_{j,i}$, $\phi_{j,i}$, and $\phi'_{j,i}, i = 1, n$ define the adjunct orientations for elliptical volumes V_i and V_i' respectively.

Similarly, for a cylinder,

$$\begin{aligned} D^i(V_i, V_i') = & \alpha \left(\frac{\min\left(\frac{L}{R}, \frac{L'}{R'}\right)}{\max\left(\frac{L}{R}, \frac{L'}{R'}\right)} \right) + \\ & \beta \left(\frac{\sum_j \theta_{i,j}\theta'_{i,j}}{\sqrt{\sum_j \theta_{j,i}\theta_{j,i}}\sqrt{\sum_j \theta'_{j,i}\theta'_{j,i}}} \right) + \\ & \gamma \left(\frac{\sum_j \phi_{i,j}\phi'_{i,j}}{\sqrt{\sum_j \phi_{j,i}\phi_{j,i}}\sqrt{\sum_j \phi'_{j,i}\phi'_{j,i}}} \right) \end{aligned} , \tag{63}$$

where L and L' are the lengths and R and R' are the radii of cylindrical volumes V_i and V_i' respectively.

In comparing a computed model against a stored model we are once again confronted with the problem of correspondence, i.e. that of determining which primitives in the computed

model correspond to which in the stored. This could be posed again as a consistent labeling problem [Hummel & Zucker 1983] in which the object of the process is to associate to each primitive in the computed model a corresponding one in the stored model that maximizes global consistency. Compatibility functions $R_{ij}(\lambda, \lambda')$ are readily defined in terms of the adjunct relations describing the structure of the local neighbourhood. Fortunately, however, we have an additional constraint in terms of organizational structure that can be used to simplify the process. The hierarchical ordering of the ellipsoid-cylinder model imposes the global structure that is the by-product of the relaxation labeling process. By matching the root notes (i.e. the largest primitive according to our definition) of two hierarchies, corresponding elements can be efficiently located with a directed search procedure.

The alternative to identification by prototype comparison is to describe an object in terms of features computed from the ellipsoid-cylinder model. Such an approach works well where features exist that precisely differentiate objects belonging to different classes of models. We used this approach in an attempt to classify an ellipsoid-cylinder model of a blood platelet into one of three classes based on simple geometric properties. Platelets are grouped into three categories on the basis of morphology [Frojmovic & Milton 1982]:

- Discocytes: platelets in their normal state. Their shape can be described as being similar to that of a torus except that a concavity replaces the hole in the center.
- Echinocytes: platelets that are ellipsoidal in shape. These sometimes have projections (pseudopods) coming out of the surface which can comprise a substantial portion of cell volume.
- Disco-Echinocytes: platelets not falling into the above two classes. As platelets undergo change, they vary in shape between discocytes and echinocytes.

A classification of an ellipsoid-cylinder model representing a blood platelet can be made by viewing platelets as oblate ellipsoids with diameter D, thickness T, and calculated axial ratio T/D. These dimensions are readily computed from parameters of the principal ellipsoid in

the model; the presence of pseudopods is verified by noting whether the principal ellipsoid has cylindrical primitives attached to it. A model is classified as a discocyte if the axial ratio T/D is less than 0.5 with no pseudopods present. Disco-echinocytes possess pseudopods and have an axial ratio that varies between 0.5 and 0.9. Echinocytes are spherically shaped with a ratio greater than 0.9 and typically possessing many pseudopods. In addition to classification, one can also obtain an estimate of cell volume and shape change over time.

5.1.4 Summary

The problem of geometric inference is defined as that of inferring the geometric structure of an object from its surfaces, and is composed of three sub-problems: primitive instantiation, primitive conjunction, and model interpretation. The process of primitive instantiation is applied to each surface patch S_i of the composite surface graph G^c. This consists of first identifying a primitive model (i.e. an ellipsoid or cylinder) that best describes the geometry of S_i based on surface differential geometry. The parameters of this model are then computed by sampling S_i and applying the appropriate fit to data. To simplify the task, S_i is first mapped into standard viewing position by determining the axes of its intrinsic coordinate system (ICS).

Once a set of primitive models has been computed from G^c, multiple instantiations arising from unresolved occlusions or errors in the reconstruction process must be eliminated. This is accomplished with a similarity measure operating on model parameters that eliminates like primitives occupying the same space in 3D. The resulting set of primitive models is then organized into a representation for the object by taking the conjunction of all primitives. The composite geometric model resulting from this process can be interpreted as an object either through comparison with a set of stored models or by computing feature descriptions of the model from which the object can be identified.

5.2 Experiments on Artificial and Real Data

One difficulty in working with shape in three dimensions is visualizing the different computational processes involved. Towards this end, we designed a computational environment[44] that enabled us to implement the ideas presented in this work and visualize the results both quantitatively and qualitatively. The system is composed of the following three modules:

(1) Surface Analysis Module: Computes a set of Dynamic Intrinsic Images for each frame in a sequence of images. The DII is the embodyment of the single view surface graph as described in Chapter 3.

(2) Motion Analysis Module: Computes the set of interframe transforms describing the movement of a solid object in 3D from the set of DII's computed above (Chapter 4).

(3) Geometric Inference Module: Reconstructs the surfaces of an object in a common reference frame from a set of DII's and their associated interframe transforms. The geometric inference paradigm is applied to the result, the composite surface graph G^c, from which an ellipsoid-cylinder model is generated (Chapter 5).

In this section we present the results of three experiments on applying the geometric inference paradigm to composite surface graphs computed from the surfaces and motion of objects in 3D. The first experiment was performed on the G^c computed from the artificial blood platelet model shown in figures 30a-30d. The second experiment was performed on the G^c computed from TV images of a real blood platelet in motion under the optics of a light microscope. Finally, the third experiment was performed on the G^c computed from TV images of the stone owl shown earlier in figure 22a.

---

[44] As yet without a name for lack of a clever acronym, but known locally as the Three-Dimensional Interpretation Testbed. .

5.2.1 Artificial Blood Platelet Model

Four views, at 0°, 96°, 192°, and 288°, of an artificial blood platelet model in rotation about the vertical axis were shown earlier in figures 30a-30d. The image sequence is made up of 30 frames, each related by a 12° rotation with a 128 × 128 pixel resolution. Four views of the reconstructed surface model corresponding to this sequence were shown earlier in figures 39a-39d, and a depiction of the composite surface graph for this sequence is shown in figures 45a and 45b. Notice that the graph is composed of 11 surface patches, with one corresponding to an ellipsoidal primitive and the remainder to cylindrical primitives. The geometric inference paradigm is then applied to G^c, yielding the results shown in figures 46a-46d.

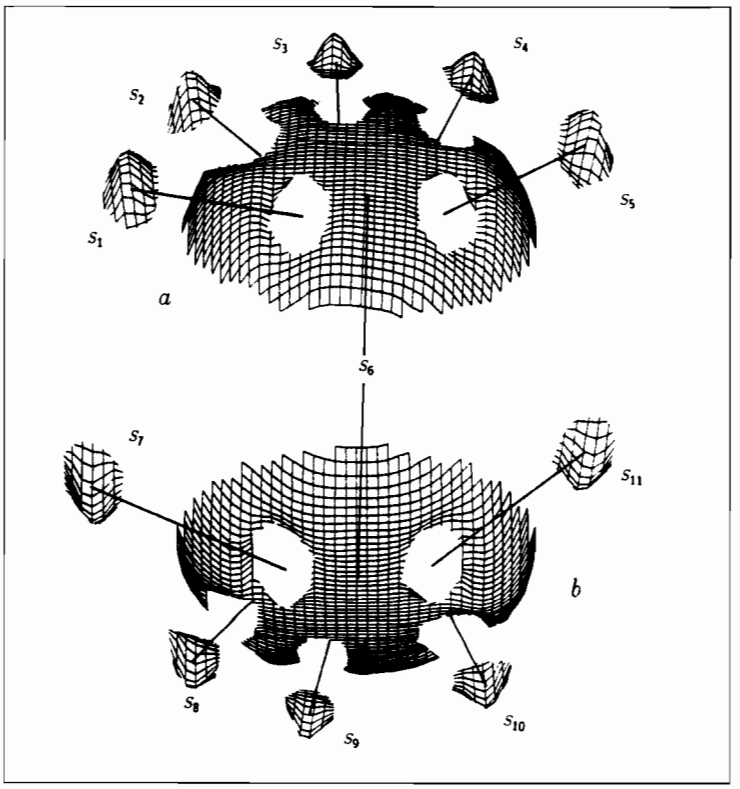

Figure 45a & 45b Composite surface graph for the blood platelet model - front and rear view surfaces

A casual glance at the result of figure 46 shows that the process correctly identified the types of primitives involved as well as their position in 3D. The process does make slight errors

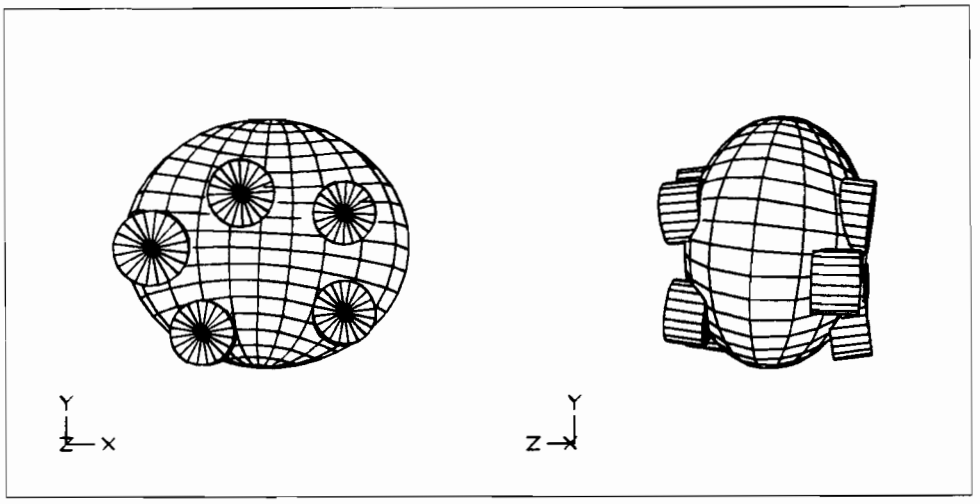

Figure 46a & 46b Ellipsoid cylinder model of the artificial blood platelet - 0 and 90 degree views

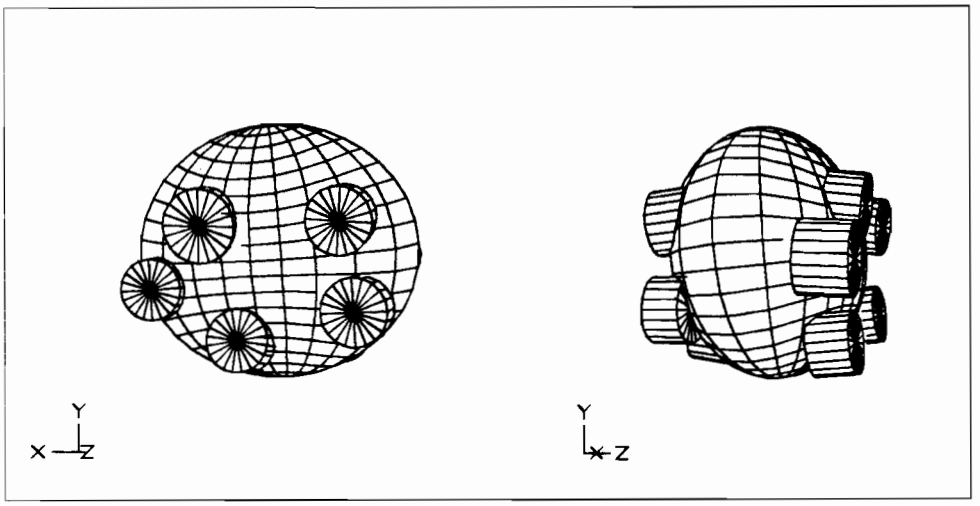

Figure 46c & 46d Ellipsoid cylinder model of the artificial blood platelet - 180 and 270 degree views

as to the orientation of the cylindrical primitives and to how they are joined to the principal ellipsoid. These are attributable to difficulty in estimating the intrinsic coordinate system (ICS) of the cylinders. The ICS is determined by computing the eigenvectors of the inertia tensor matrix, as described in Chapter 2, where the inertia tensor is estimated from samples of the surface patch. Estimation error in the present case can be traced to the following causes:

- The surface sampled is actually a cone and not a cylinder; distortion of the surface in the vicinity of the peak tends to shift placement of the axis of symmetry.

- The number of surface samples is small and distorted by noise, thus increasing the uncertainty of axis positions.

Errors in the join between the cylinder and ellipsoid surfaces are partially due to errors in axis placement, but are more dependent on errors in localizing the boundary between the two surfaces. Recall from Chapter 3 that surfaces are decomposed on the basis of extrema in surface normal curvature. In the result shown, the boundary is displaced by one pixel towards the cylindrical surface patch. This effectively shortens the cylinder such that it sits in mid-air along the periphery of the cylinder. The corresponding join on the surface of the ellipsoid (which has the appearance of a moon crater) is effectively smoothed out when its surface is parameterized.

In spite of these errors, the result is quite satisfactory. Note, in particular, that the relative dimensions of the primitive models fit to the surface patches, as they are quite faithful to the original surface. This experiment shows that, at least in theory, the geometric inference paradigm is a correct approach to the problem of determining the geometry of an object from its surfaces for restricted classes of objects. However, the artificial platelet model is unsatisfactory from the viewpoint that the data was idealized (i.e. the model was itself composed of primitives similar to those used to model the object). The next two experiments address this issue by dealing with real, non-ideal data composed of entirely smooth surfaces presented to the system as sequences of TV image frames.

5.2.2 Real Platelet Data

Figures 47a and 47b show two images of blood platelet photographed from the video monitor of a light microscopy system. The system was composed of a Zeiss Universal microscope with a DIC objective using the Allan video-enhanced contrast microscopy technique. The total magnification from cell to video screen was approximately 14.7 (roughly 0.2 microns) with illumination supplied by an incandescent bulb at 10 volts. Images were input with a

Hamamatsu model C1000 (Type 01) camera and recorded on a Panasonic $1/2''$ video cassette unit. The resolution of the recorded images is approximately 256×256, of which a quarter or a 128×128 sub-image is shown in figures 47a and 47b. A particular problem with this set-up is the relatively low resolution of the resulting images. The platelet shown in the figures is not typical of the data, and is in fact a giant that comprises a very small percentage of the total population. Typical platelet sizes are about one third to one quarter the size of the one shown. It is difficult to attain higher magnifications as the current setting is near the physical limits of light microscopy.

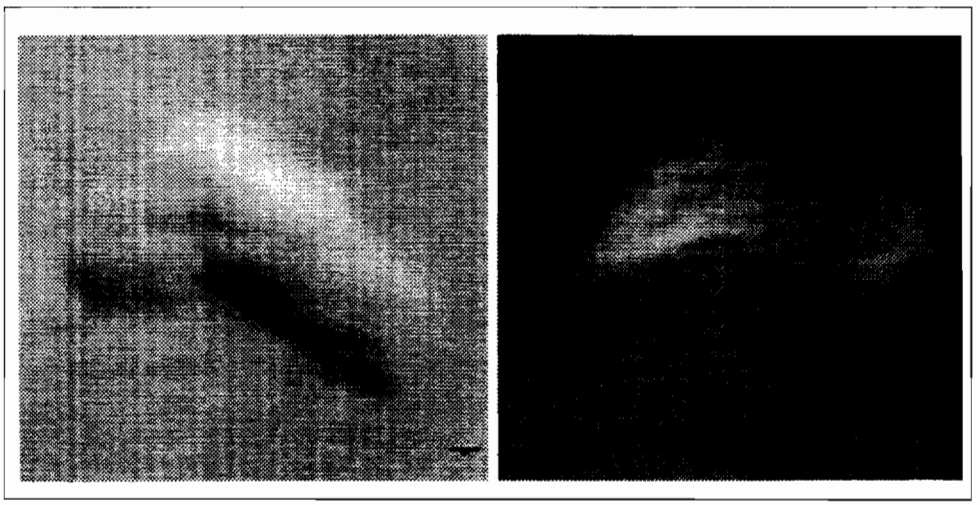

Figure 47a & 47b Real blood platelet image - front and rear views

Another problem with the set-up was the difficulty in recording platelet motion, particularly that of capturing the cells in rotation. The cells had a tendency to flip over rapidly in comparison to their slow translational velocity, in effect violating the assumption of continuous motion (i.e. no instantaneous change in velocity). For this reason we simply took two photographs showing the platelet in front and rear views and relied on the system's use of characteristic view constraints for correct reconstruction of the two surfaces. However, this method failed because the cell did not remain rigid during the course of its rotation. The only alternative was to attempt geometric inference from a single view under the assumption that the cell was symmetric (it was). Figures 48a and 48b show the surfaces recovered from

the corresponding images in figures 47a and 47b, and figures 49a to 49d show the result of applying the geometric inference paradigm to the surface graphs computed from each image. An interesting aspect of this experiment is the decomposition of each surface into the two patches from which the resulting cylinders were derived.

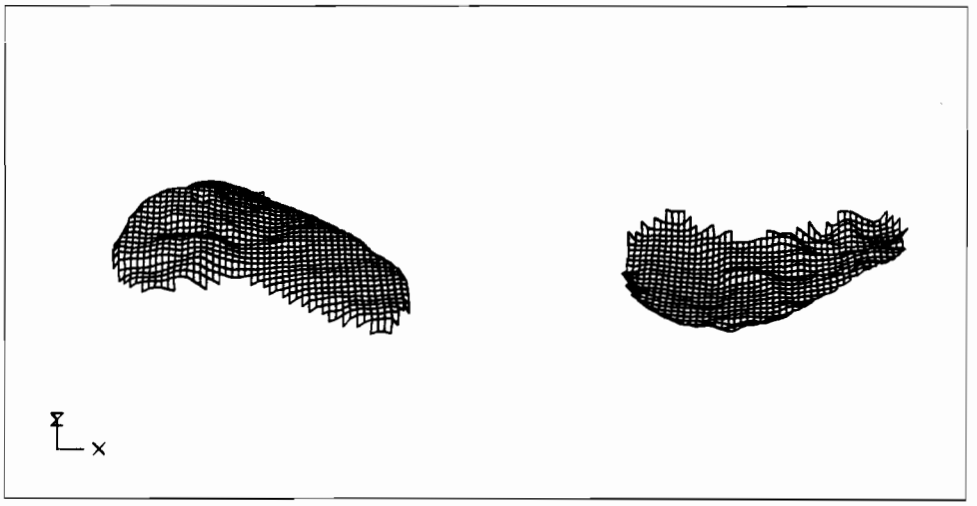

Figure 48a & 48b Reconstructed surfaces corresponding to the images in figures 47a & 47b

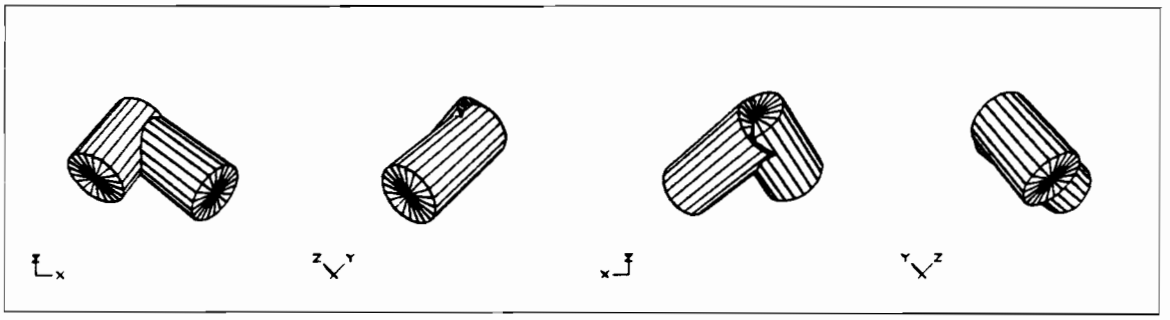

Figure 49a – 49d Ellipsoid-cylinder model of the real blood platelet - 0, 90, 180, and 270 degree views

The surfaces of the platelet are smooth and do not give rise to extremal values in surface curvature, thus the surface should be viewed as a single entity. However, the occluding contour of the surface definitely suggests the breakup of the surface into two sections by a segment whose endpoints lie on peaks in the curvature of the occluding contour [Marr 1982]. This would be more apparent if it were possible to view the platelet "edge on", i.e. by a

90° rotation about its longitudinal axis. Because we do not have access to the complete surfaces of the object in this case, it becomes necessary to make inferences on the basis of additional cues such as provided by occluding contour. Dill [1986] made use of constraints on boundary contour to decompose the surfaces of a cell into sections for the purpose of analyzing morphology. The result shown was obtained by augmenting the geometric inference procedure described earlier to include similar boundary decomposition cues. In this example, the surface was decomposed into two patches along the segment running between the two peaks in boundary curvature. We have not investigated how such information can be used in general, but it is evident that the geometric inference paradigm requires additional constraints when not all of the surfaces of an object are accessible.

We were not successful in attempts to analyze the shape of more typical platelets using the inference paradigm as originally described. Part of the reason has to do with the low resolutions involved as the reconstructed surfaces are effectively low pass filtered, revealing little detail of the cell surface. The other part of the reason has to do with cell motion, i.e. the fact that cells do not remain rigid during the interframe interval. In order for our approach to become feasible for this application, image resolutions must be increased and the motion of cells must be such that they are at least locally rigid in time. The results obtained, however, at least demonstrate the possibility of applying the shape model to problems of biomedical image analysis. While the platelet application was not as successful as hoped, the results obtained under very poor conditions suggest that this method might be successful for problems of a similar nature where conditions are more favourable.

5.2.3 The Owl Revisited

As a final application of the geometric inference paradigm, we return to the earlier example of the stone owl statuette rotating and translating in front of a TV camera. The owl was

illuminated by two light sources[45] situated approximately three inches above the camera at a distance of 10 feet from the scene. The camera was a Hitachi model KP-120u with a resolution of approximately 320×256 and equipped with a 15-40mm variable zoom lens. Images were scaled to an even aspect ratio and averaged down to a 128×128 resolution. To avoid problems with uneven reflectance and specularity, the owl was coated with a mat finish.

Four views of the owl at approximately 0°, 90°, 180°, and 270° were shown earlier in figures 37a-37d respectively. The geometric inference paradigm was applied to the surface graph computed for these images with the results shown in figures 50a-50d, at approximately the same viewing positions as above. It is interesting to note that the resulting model is composed entirely of 6 cylinders with one corresponding to the head, one to the torso, two for the wings, and two for the eyes, which are just barely visible. The reason for this is that the algorithm failed to correctly compute the orientations of the two cylinders corresponding to the eyes, which should be normal to the plane of the viewer. Again, as in the case of the blood platelet model, it is difficult to correctly estimate the ICS of primitive models when the number of surface samples is small.

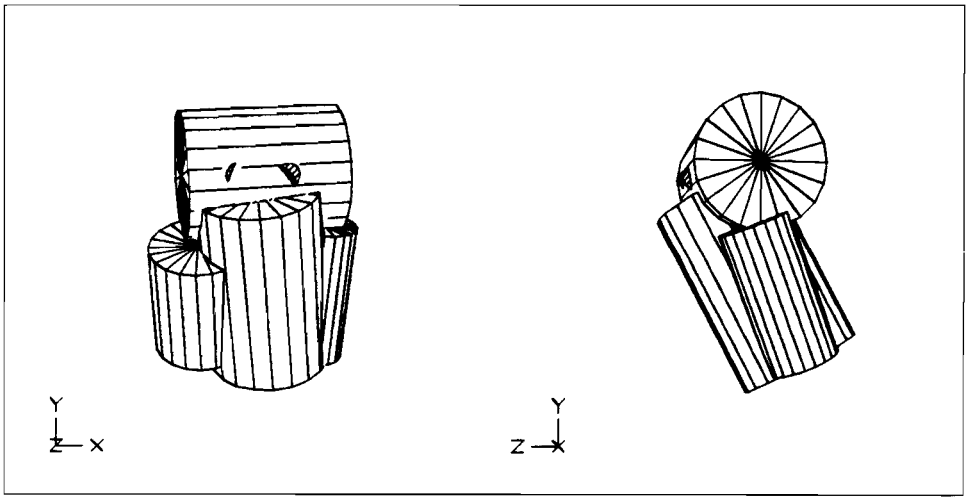

Figure 50a & 50b Ellipsoid-cylinder model of the owl - 0 and 90 degree views

[45] An Intralux 5000 illuminator consisting of a 185 watt light source directed to two lenses by two flexible fiber-optic bundles.

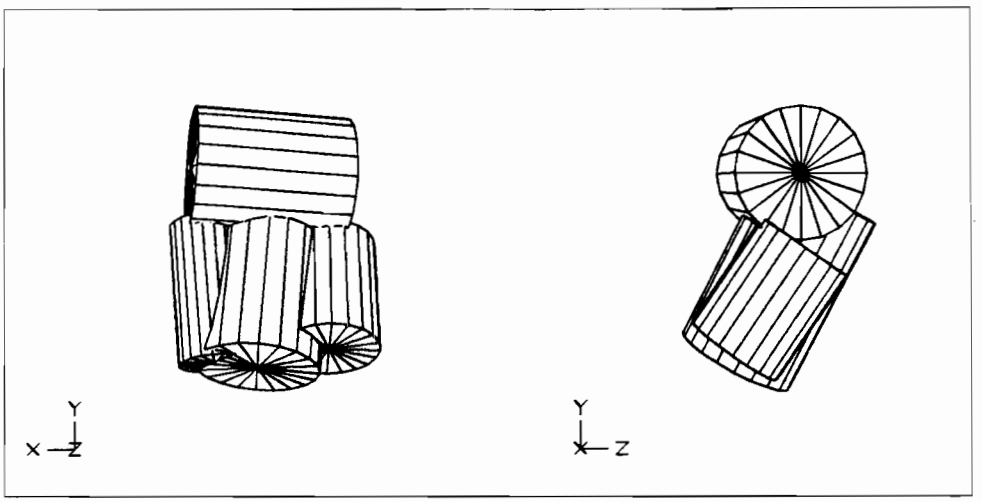

Figure 50c & 50d Ellipsoid-cylinder model of the owl - 180 and 270 degree views

The composite surface graph for this object actually consists of 4 surface patches since the head and wings of the owl are joined together by the surface in the rear view. However, the effect of merging these patches would be to obscure the features of the owl into a single cylinder that hides all remaining details. In cases such as this one, it is not clear whether or not to include in G^c a surface patch that overly simplifies the interpretation of the object when it is added to the composite model. A reasonable strategy to employ in this situation would be to apply Marr's principle of least commitment [Marr 1982] and provide both interpretations to higher levels of processing. The result shown in figures 50a-50d is the more "interesting" of the two interpretations, and is a reasonable interpretation of the geometry of the owl given the limited vocabulary of ellipsoids and cylinders.

5.3 Chapter Summary

The final component of the shape problem defined in Chapter 1 is the abstraction of the reconstructed surfaces of an object as a composite geometric model composed of primitives such as ellipsoids and cylinders. This task, referred to as the problem of geometric inference, is based on an interpretation of the surfaces of an object in terms of properties of surface differential geometry. In this chapter we showed how a representation that makes such properties

explicit, the composite surface graph G^c, could be used as the basis for a process in which the local geometric structure of an object is inferred from the curvature properties of its surface. The constraint upon which this inference is based is a property of differential geometry in the large which states that the global structure of an object can be deduced from its surfaces, provided that these surfaces are homogeneous in certain properties. G^c meets this requirement be representing the surface as a set of homogeneous patches formed by the decomposition of the surface along extrema in surface normal curvature. The principal curvatures of each patch are used to select the most appropriate geometric primitive as a representative of the local geometry of the patch, from which the local model is parameterized. The conjunction of all resulting primitives is used as a composite geometric model for the underlying object in a scene.

A set of experiments was performed on three sets of data: images of an artificial model of a blood platelet in motion, images of a real blood platelet in motion as seen through the optics of a light microscope, and images of a stone owl statuette undergoing rotation and translation in front of a TV camera. In each case, the geometric inference paradigm was able to provide a fair rendition of the object according to the primitive model vocabulary available. However, the experiments also indicated that the paradigm should be extended to take into account constraints provided by occluding contour (as in the case of the real blood platelet data). Another aspect of the problem, revealed in the third experiment with the owl data, concerns objects that give the appearance of different symmetries as a function of viewpoint (e.g. the owl viewed from the rear appears as a single cylinder, whereas the front view gives an entirely different appearance). It is obvious that the geometric inference paradigm covers but a subset of the methods by which object geometry is inferred from surfaces. Yet in spite of its limited nature, the paradigm was capable of capturing the essential geometric characteristics of the data presented to it, and should also serve as the basis for future research.

Chapter 6 Concluding Chapter

In this final chapter, we summarize the basic concepts of the shape model put forward in this thesis, the results obtained with this model, and what we view as the principal contributions of this work. The chapter concludes with some suggestions as to a course of future research based on the shape model.

6.1 3D Shape in Motion - Summary

The basic purpose of this work was to address a number of fundamental questions regarding the computation of three-dimensional shape. Our approach to the problem consisted of proposing, and then exploring, a framework for shape in which three-dimensional descriptions of an object are computed from a sequence of images of the object in motion. These images can portray the object in motion about the viewer, or vice-versa, but no a priori relationship is assumed to exist among them. We began our investigation by adopting the premise that the basis for shape is contained in the surfaces of an object. The first question dealt with the nature of shape representation. This led to the consideration of how surfaces can be interpreted as objects, in particular how the local structure of a surface can give rise to geometric interpretations. The result was a hierarchical representation for shape at two primary levels. The first was a representation for the surfaces of an object, defined locally in terms of pointwise feature descriptions, and more globally in terms of regions or patches composed

of these feature descriptions. The second level of representation focused on the geometric characteristics implied by the surface representation, locally in terms of volumetric primitives (ellipsoids and cylinders), and globally as a conjunction of primitives along the lines of Marr's [1982] generalized cylinder approach.

Consideration of how descriptions based on this representation were computed and the constraints that make it possible led to a decomposition of the computational problem as a set of three sub-problems to be solved in succession:

(1) The location of surfaces from different viewpoints and the computation of their intrinsic properties as defined in terms of differential geometry.

(2) The estimation of object (or viewer) motion and the reconstruction of surfaces in a world coordinate system.

(3) The computation of a composite geometric model and its subsequent interpretation from the above surfaces and their intrinsic features.

This problem decomposition followed quite naturally from our initial premise regarding the role of surfaces in shape perception. In order to solve the second and third sub-problems, it was first necessary to make those aspects of surface structure explicit (i.e. the constraints) that enabled solution of the correspondence problem implied by (2) and the inference problem implied by (3). The second sub-problem was a consequence of the fact that non-symmetrical objects require information from multiple viewpoints. Its solution (i.e. determining the relationship among the different viewpoints) enabled the reconstruction of object surfaces and their descriptions in a common coordinate frame. Finally, the third sub-problem was concerned with the transition from this surface-based description to a volumetric model comprised of ellipsoids and cylinders.

A framework for shape and computational approach to its implementation were outlined in Chapter 2. In order to be able to address questions of "top-down" feedback and embedded

knowledge, we chose a domain in which the constraints were relatively well-understood, that of piecewise smooth, solid objects. The motivation for the computational approach was a constraint from differential geometry *in the large* [Hilbert & Cohn-Vossen 1952] which stated that global inferences about geometric structure could be made on the basis of local observations, provided that continuity and homogeneity requirements were met. For this reason, we introduced a representation for surfaces called a surface graph, in which the surfaces of an object were decomposed into a set of continuous surface patches, homogeneous in the signs of principal curvatures. These patches formed the basis of what we called the geometric inference paradigm, where each patch on a surface was assumed to correspond to a volumetric primitive. Patch curvature characteristics were used to select the most appropriate volumetric primitive, whose parameters were subsequently determined from the points comprising the patch.

Another function of the surface graph was to make explicit intrinsic properties of the surface that could be used in computing correspondence across adjacent viewpoints. Principal curvatures and directions were the essential properties computed for this purpose, with their extremal values serving as tokens in the matching process. Correspondence was computed by correlating the structure of two neighbourhoods in terms of these intrinsic properties, from which two sets of corresponding surface points were obtained. Because of an assumption of rigid-body motion, these point correspondences were subsequently used to estimate the components of a linear transformation describing the motion of the object through two adjacent views. By concatenating successive transformations, it was then possible to reconstruct the surfaces, and by extension the surface graph, of an object in a common coordinate frame. The result of this process was the composite surface graph G^c, a representation for the complete surfaces of an object as presented to the viewer in a sequence of views.

The remaining component of the framework outlined in Chapter 2 focused on the geometric interpretation of the surfaces of an object. Our response to this problem was the geometric inference paradigm, as applied to the composite surface graph G^c. The idea here was that if

an object could be idealized as a conjunction of primitive geometric components, then it should be possible to derive these components locally from surface properties by applying constraints from differential geometry. Of these constraints, the most important was the observation that the intersection of geometric primitives resulted in a marked change in curvature along the points of intersection. By decomposing a surface along contours containing these points, the resulting patches maintained a correspondence to no more than a single geometric primitive. It was then straightforward to identify the primitive associated with each patch on the basis of the signs of its principal curvatures. A parametric description could subsequently be obtained by fitting the appropriate model to points sampled from the patch.

Chapters 3 to 5 filled in the outline of this framework, with each chapter corresponding to each of the three sub-problems that defined the shape problem. In Chapter 3 we showed that the computation of the surface graph from a single viewpoint was equivalent to computing a set of intrinsic images describing features of the surface. We called these Static Intrinsic Images to differentiate them from the aggregate set of intrinsic images computed over a sequence of images. We referred to the latter as Dynamic Intrinsic Images. The procedure to compute the intrinsic images consisted of two parts, the location of points describing the surface of an object in an image followed by a local parameterization of the surface from which differential properties were computed. Although the framework did not address the problem of surface location, our implementation used a shape-from-shading technique to obtain the image depth map, the basis for subsequent computation. We defined the notion of a "curved edge" based on extremal values of surface normal curvature. Contours formed by these distinguished points were used to decompose the surface into the set of homogeneous surface patches that defined the surface graph. In addition, the set of intrinsic image features at each point on a surface patch defined a vector describing the characteristics of the surface on a pointwise basis.

The integration of surface descriptions from multiple views was the topic of Chapter 4. We showed that the same features used to decompose the surface into constituent patches, i.e. "curved edges", could also serve as match tokens for a correspondence process. The approach

taken was a correlation of local neighbourhood structure, where structure was defined in terms of the set of intrinsic feature vectors defining the immediate neighbourhood of a token point. By using this notion of structure, we were able to minimize the ambiguity of the matching process as well as reject point correspondences of questionable certainty. Another aspect of this approach was the use of the mean value of match correlation values (i.e. match affinities) as a measure of global correspondence. This statistic was used in deciding whether or not correspondence could be computed on the basis of local measures, such as in cases of occlusion. In cases where it was not possible to maintain correspondence locally, a global approach based on matching the structure of occluding contours was invoked in an effort to re-establish correspondence. The solution of the correspondence problem permitted the estimation of the linear transformations describing object motion during the interframe interval. These interframe transforms enabled the computation of a composite transform that was attached to each dynamic intrinsic image. This composite transform in turn provided the mapping by which each set of DII's could be transformed into a common coordinate system, thus establishing a dual frame of reference. In doing so, overlapping surfaces and descriptions were eliminated, resulting in a unique set of surfaces as well as the composite surface graph G^c.

The implementation of the geometric inference paradigm was the topic of Chapter 5, and provided an opportunity for exploring the competence of the framework as a whole through experiments on real and artificial data. The paradigm itself was presented as a set of three sub-problems to be solved in succession. They were the instantiation of each surface patch of G^c as a primitive geometric model, the determination of the parameters for each model and the organization of all instantiated primitives into a composite model, and lastly, the interpretation of the composite model as an object in a domain. Primitive models were instantiated on the basis of the signs and magnitudes of principal curvatures. A particularly useful statistic was the mean ratio of principal curvatures computed over the surface of a patch. This was used as a similarity measure in selecting a best representative primitive model in cases where surface

geometry was ambiguous. Once a particular model was selected as a representative for a patch, its parameters were determined in two step process. First, the axes of the model were computed from an estimate of its inertia tensor, obtained from points comprising the surface patch. This allowed the surface to be rotated and translated into a standard viewing position from which the parameters of a canonical model could be determined as the second step in the process. The resulting primitives were then organized into a composite model in hierarchical fashion along the lines suggested by Marr [1982]. The last step in the process was that of interpreting the result, either by comparison to a prototype, or on a feature basis. Our work did not investigate this last problem in detail, but did consider it in the context of a problem in blood platelet morphology.

6.2 Implementation Results

The results obtained in the implementation of the shape framework fall into two categories: those having to do with the framework as a whole, and those representing specific instances of problems in computer vision. We will deal with each of these separately.

6.2.1 The Framework as a Whole

One of the objectives of this research was to be able to compute a hierarchical description of the shape of an object at the level of surfaces, and at the level of the geometry implied by these surfaces. Towards this end, the experiments in Chapter 5 confirmed that the computational model outlined in this work was successful in this regard. The experiment with the artificial blood platelet model was quite instructive because it demonstrated the invertibility of the shape paradigm. That is, starting with a geometric model satisfying all the assumptions of the paradigm, and creating a sequence of images representing this model in motion, the process was able to correctly recover its 3D structure from the sequence of 2D images.

The next escalation from our basic goal was the application of the same computational model to a real object in order to obtain an idealized representation of the object in terms of ellipsoids and cylinders. Results obtained from images of a stone owl statuette in motion showed that this was indeed possible. Furthermore these results were obtained in a practical viewing environment, i.e. using a low resolution television image illuminated with a single light source at the camera, at a distance of approximately 10 feet from the object. However, this experiment also pointed out that the simple approach of the geometric inference paradigm will not always lead to a correct representation, as was evidenced by the conflicting front and rear views of the owl. It is in cases such as this one that higher level knowledge must come into play in order to provide the necessary additional constraints, a topic for future research.

One of the original goals of the project was to develop a computer vision system capable of interpreting blood platelet morphology from observations of blood cells in motion under the objective of a microscope. We were able to demonstrate the feasibility of such a system in terms of reconstructing cell surfaces and obtaining elementary renditions of idealized cell geometry. However, we were unable to obtain detailed models of cell morphology, primarily because of the low image resolutions involved[46]. Yet our results did demonstrate the applicability of the shape framework to other problems in cell biology where sufficient resolution is available.

6.2.2 Specific Problems in Computer Vision

In the course of developing our model for shape, we were confronted with a number of problems fundamental to computer vision. The first one of these was that of determining the 3D shape of an object from its image using shape-from-shading. We demonstrated that although exact recovery of shape from shading was not possible for elliptical and hyperbolic

---

[46] Typical cell sizes were on the order of 64×64 pixels, insufficient for the algorithms employed in our implementation

surfaces, reasonably precise recovery was possible if certain constraints regarding surface curvature were met. The conditions under which this was possible and the errors involved were derived for the case of a Lambertian reflectance function. Furthermore, the practicability of using shape-from-shading was demonstrated as all experiments presented in this work were based on data obtained with this method.

The problem of surface representation and the computation of surface descriptions occupied much of Chapter 3. Using the output of the above shape-from-shading algorithm, we were able compute depth maps of real and artificial surfaces of moderate complexity, from which piecewise-continuous approximations to surfaces were obtained. This made it possible to compute the features of surface curvature required for the derivation of the surface graph as a set of intrinsic images. The principal result of Chapter 3 was the demonstration that the surface graph representation for a surface could be computed from the image of a real object.

In Chapter 4 we considered how to compute a composite surface graph by integrating information from different viewpoints. This meant the solution of a difficult correspondence problem, given the smooth nature of the surfaces in our examples. We analyzed the structural correlation model on intensity and range images of an artificial model undergoing known translation and rotation. This approach was able to maintain almost exact correspondence where sufficient features were present. However, the more important result was that we were able to predict when motion parameters obtained from this process were invalid without any knowledge of the object or its motion. We also demonstrated how correspondence could be re-established by using global features such as occluding contour when local processes were unable to maintain correspondence. The combination of local and global processes enabled the reconstruction of the surfaces of an object from different viewpoints, as was shown with the surfaces of the owl and the artificial blood platelet model.

Most of Chapter 5 was devoted to overall results obtained with our model for shape as discussed in the previous section. However, we also observed an interesting problem

concerning the parameterization of an object from partial views of its surfaces. This was the occurrence of parameter and displacement error when trying to fit a model to samples obtained from a partial view of a surface. We found that this problem could be resolved in one of two ways: by including data samples from other viewpoints, or the inclusion of symmetry assumptions in the parameterization process. An example of the second approach was the use of a parallelepiped approximation in the parameterization of cylindrical surfaces as right cylinders.

6.3 Contributions

Our work is in many ways a synthesis of ideas and concepts that have been around computer vision for the past decade. However, most of these have been looked at only in isolation, or else together in theoretical frameworks that have received little in the way of practical evaluation. Part of the reason for this, of course, has been the complexity involved in implementing such frameworks. The motivation for our work originally stemmed from an application in biological image analysis, that of quantifying and interpreting the shape of blob-like objects such as blood cells using simple geometric primitives. In researching this problem, a microcosm of the general vision problem, it was necessary not only to address the individual components (i.e. from images to surfaces to objects), but also to tie them together in computing a hierarchical representation for shape. This meant identifying the constraints by which one level of representation was computed from another, spanning the range from local to global. In doing so, we were able to develop a simple, yet powerful, computational model for shape that can cope with a large variety of objects. Furthermore, the implementation of this model was important in providing a general insight into fundamental problems of vision, much like the role of experimental physics. We view this as the principal contribution of our work.

This principal contribution is supported by the following contributions:

- The development of a representation for surfaces, called a surface graph, and its associated computational model. We showed how this representation could be exploited in obtaining solutions to the correspondence and geometric inference problems. Moreover, the surface graph representation also demonstrated the validity of the intrinsic image concept. By augmenting intrinsic images with a viewer-to-world transformation matrix computed from motion correspondence, we were able to implement the composite surface graph G^c as an equivalent set of what we called *dynamic intrinsic images*. This made it possible to refer to surface features in both viewer-centered and world coordinates, a pre-requisite to the geometric inference paradigm.

- We showed that elliptical and hyperbolic surfaces could be recovered to an acceptable degree of accuracy using a shape-from-shading approach [Pentland 1982a,b]. By analyzing the effect of error components on the surface integral, we were able to show that the error in the resulting depth map was acceptable for elliptical and hyperbolic surfaces with eccentricity ratios as high as 5:1.

- A model for correspondence based on the correlation of the local structure of the surface graph was developed and shown to be capable of accurately recovering the motion parameters of objects with curved surfaces. More importantly, the model was able to detect those cases in which correspondence could not be established, i.e. where the local correspondence model broke down.

- Experiments with the correspondence model showed that correspondence based solely on tokens derived from intensity features is insufficient for smooth surfaces because of the difficulty of localizing such features.

- The detection of a breakdown in the local correspondence model was used to delimit segments of "continuous" motion, i.e. where the local model was successful. We showed that such segments could be spliced together on the basis of characteristic views (in our case, by exploiting constraints imposed by the occluding contour of an object). This made it possible to compute for each view of an object, a transformation mapping viewer-

centered coordinates into a world reference frame, which permitted the integration of surface descriptions from different viewpoints.

- We addressed the problem of interpreting local surface geometry by introducing the geometric inference paradigm, and developed a computational model for solving this problem in a domain of objects composed of ellipsoids and cylinders. We presented experimental results in which ellipsoid-cylinder representations of objects were automatically generated from sequences of images of those objects in motion.
- A test bed for experimenting with three-dimensional representations was developed in which our model for shape was implemented. This is important because it demonstrates the practicability of the ideas presented in this thesis for solving real problems in computer vision. It also permits a detailed analysis of the behavior of algorithms from which further insight into the nature of 3D shape problems can be gained.

6.4 Further Work

The intent of this research was to develop a computational model for shape in a domain in which objects could be represented as models composed of ellipsoids and cylinders. To the extent of our initial objectives, the results have been generally successful, and suggest possibilities for future research. Two areas in particular deserve further attention. The first concerns the kinds of model primitives that are appropriate for three-dimensional shape representation, and the second deals with the task of how such models are interpreted as objects.

The choice of model primitives is largely a compromise between what is required to adequately reflect important features of an object and what can be computed from an image or measured from a scene. For example, Marr's [1982] choice of generalized cylinder models is related to their competence in representing many different forms. In addition, constraints imposed by symmetry and occluding contour form the basis of a computational model by which cylindrical primitives can be located in an image. Our choice of ellipsoid and cylinder

primitives has a similar motivation. We are able to represent a wide variety of objects using such primitives and have the necessary constraints from differential geometry which allow their extraction from an object's surface description. However, these representations are still quite limited when considering more complex forms such as found in nature.

As an alternative to fitting scene parameters to a fixed set of primitives, one might instead opt for an approach where an elastic primitive is deformed into correspondence with scene data. The analogy is that of a sculptor who molds a lump of clay into a likeness of some person or object. In fact, such an approach has already been suggested [Witkin and Tenenbaum 1984] and some preliminary results are reported in Pentland [1985]. Pentland shows quite convincingly how a representation composed of superquadric primitives can be used to synthesize a wide variety of natural objects. What is most appealing about these primitives, however, is the fact that their parameters are overconstrained in terms of measurements of surface tilt. This makes it possible to obtain reliable estimates of surface shape in terms of superquadrics [Pentland 1985]. An approach to shape representation based on elastic primitives would be a natural extension of the geometric inference paradigm, and for this reason should be investigated further.

Another problem that must be dealt with is that of resolving conflicting interpretations in the geometric inference paradigm. Recall that in computing the ellipsoid-cylinder model of the owl, we were confronted with a situation in which the inclusion of rear-view surface patches eliminated much of the structure derived from the other viewpoints. The reason for this was that the cylindrical appearance of the rear view led to the inference of a cylinder that completely enveloped the object, i.e. a false assumption of symmetry. Our solution was simply to ignore this interpretation on the basis that it eliminated much of the structure of the object, but we could have easily encountered a situation in which the choice was not as simple. It is at this point that the "bottom up" course we have been pursuing must give way to higher level cognitive processes that know about the structure of objects and how humans relate to them. Having knowledge about the appearance of objects in an environment is one

way in which conflicts in interpretation can be resolved.

Just how this and other problems in computer vision will eventually be resolved is uncertain. Our approach, like many others, was motivated by the Gibsonean paradigm of decoding the three-dimensional information encoded in two-dimensional images, proceeding from images to surfaces to objects. What we have attempted to do in our work is to organize many of the concepts and ideas put forward in the past decade into a framework competent for solving some applied problems in 3D shape measurement and interpretation. The hope was that by focusing on an application we could learn more about the general problem, in particular the identification of those constraints which make solutions possible. We believe that our work has been successful in this regard and hope that it will be of use to others working in the field.

References

Agin, G.J., and Binford, T.O., [1976], "Computer Description of Curved Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. C-25, No. 4, pp. 439-449.

Badler, N., and Bjacsy, R., [1978], "Three Dimensional Representations for Computer Graphics and Computer Vision," *Computer Graphics*, Vol. 12, pp. 153-160.

Barrow, H.G., and Tenenbaum, J.M., [1978], "Recovering Intrinsic Scene Characteristics From Images," *Computer Vision Systems*, A. Hanson and E. Riseman (eds.), Academic Press, New York, New York, pp. 3-26.

Bhanu, B., [1984], "Representation and Shape Matching of 3D Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 3, pp. 340-351.

Binford, T.O., [1971], "Visual Perception by Computer," *IEEE Conference on Systems and Control*, Miami.

Binford, T.O., [1982], "Survey of Model-Based Image Analysis," *International Journal of Robotics Research*, **1**, No. 1, pp. 18-64.

Clocksin, W.F., [1980], "Perception of Surface Slant and Edge Labels from Optical Flow: A Computational Approach," *Perception*, 9, pp. 253-269.

Clowes, M.B., [1971], "On Seeing Things," *Artificial Intelligence*, Vol. 2, No. 1., pp. 79-116.

Courant, R., and Hilbert, D., [1953], *Methods of Mathematical Physics*, Vol. I, Interscience, London.

Dill, A.R., and Levine, M.D., [1985], "Non-Rigid Body Motion," *Graphics Interface '85*, Montréal, Québec, May 19-22.

Dill, A.R., [1986], "Expert System for the Analysis of Cell Locomotion" M.Eng. Thesis (in preparation), Dept. Elect. Eng., McGill University, Montréal, Canada.

do Carmo, M.P., [1976], "Differential Geometry of Curves and Surfaces," Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Dreschler, L.S., and Nagel, H.H., [1982], "On the Selection of Critical Points and Local Curvature Extrema of Region Boundaries for Interframe Matching," *PROC ICPR*, Munich, Germany, October 19-22, pp. 542-544.

Duda, R.O., Nitzan, D., and Barret, P. [1979], "Use of Range and Reflectance Data to Find Planar Surface Regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-1, No. 3, pp. 259-271.

Fang, J.Q., and Huang, T.S., [1983a], "Solving Three Dimensional Small-Rotation Motion Equations," *PROC. CVaPR*, Washington, D.C., June 19-23, pp. 253-258.

Fang, J.Q., and Huang, T.S., [1983b], "Estimating 3-D Movements of a Rigid Object: Experimental Results," *PROC. IJCAI-83*, pp. 1033-1037.

Fang, J.Q., and Huang, T.S., [1984], "Some Experiments on Estimating the 3-D Motion Parameters of a Rigid Body from Two Consecutive Image Frames," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. Pami-6, No. 5, pp. 545-554.

Fischler, M.A., and Barret, P., [1980], "An Iconic Transform For Sketch Completion and Shape Abstraction," *Computer Graphics and Image Processing*, Vol. 13, pp. 334-360.

Fowles, G.R., [1970], "Analytical Mechanics," Holt, Rinehart and Winston, Inc., New York.

Freeman, H., [1978], "Shape Description Via the Use of Critical Points," *Pattern Recognition*, Vol. 10, No. 3, pp. 159-166.

Freeman, H., [1980], "Lines, Curves, and the Characterization of Shape," Report No. IPL-TR-80-004, Image Processing Laboratory, Electrical and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, New York.

Frojmovic, M.M., and Milton, J.G., [1982], "Human Platelet Size and Shape in Health and Disease," *Physiological Reviews*, Vol. 62, pp. 185-261.

Gibson, J.J., [1950], "The Perception of the Visual World," Houghton Mifflin Company, New York.

Grimson, W.E.L., [1979], "Differential Geometry, Surface Patches, and Convergence Methods,", MIT AI Memo 510, Cambridge, MA.

Grimson, W.E.L., [1981], "From Images to Surfaces," MIT Press, Cambridge, MA.

Gupta, K., [1986], "Robot Trajectory Planning in Time-varying Environments," Ph.D. Thesis (in preparation), Dept. Elect. Eng., McGill University, Montréal, Canada.

Hall, E.L., Tio, J.B., McPherson, C.A., and Sadjadi, F.A., [1982], "Measuring Curved Surfaces for Robot Vision," *IEEE Computer*, Vol. 15, No. 12, pp. 42-54.

Hilbert, D., and Cohn-Vossen, S., [1952], "Geometry and the Imagination," Chelsea, New York.

Hildreth, E.C., [1983], "The Measurement of Visual Motion", Ph.D thesis, Dept. Elect. Eng. and Comp. Sci., MIT, Cambridge MA.

Hoffman, D.D., [1980], "Inferring shape from motion," MIT AI Memo 536, Cambridge, MA.

Hoffman, D.D., [1982], "Interpreting Time-Varying Images: The Planarity Assumption," PROC. Workshop on Computer Vision, Rindge, N.H., pp. 92-94.

Horn, B.K.P., [1970], "Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object From One View," MIT MAC TR-79, Cambridge, MA.

Horn, B.K.P., [1975], "Obtaining Shape from Shading Information," *The Psychology of Computer Vision*, P.H. Winston (ed.), McGraw-Hill, New York, pp. 115-155.

Horn, B.K.P., [1977], "Understanding Image Intensities," *Artificial Intelligence*, No. 8, pp. 201-231.

Horn, B.K.P., [1979], "Sequins and Quills - representations for surface topography," MIT AI Memo 536, Cambridge, MA.

Hubschman, H., and Zucker, S.W., [1981] "Frame-to-frame coherance and the hidden surface computation: Constraints for a convex world," *Computer Graphics*, 15(3), pp 45-54.

Huffman, D.A., [1971], "Impossible Objects as Nonsense Sentences," *Machine Intelligence*, Meltzer and Mitchie (eds.), Vol. 6, pp. 295-323.

Huffman, D.A., [1976], "Curvature and Creases: A Primer on Paper," *IEEE Trans. Comp.*, Vol. C-25, No. 10.

Hummel, R.A., and Zucker, S.W., [1983], "On the foundations of relaxation labeling processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5, pp 267-287.

Hummel, R.A., Kimia, B., and Zucker, S.W., [1984], "Deblurring gaussian blur," *Computer Vision, Graphics, and Image Processing*, to appear.

Ikeuchi, Katsushi, [1980] "Numerical shape from shading and occluding contours," MIT AI Memo 566, Cambridge, MA.

Ikeuchi, Katsushi, [1983], "Constructing a depth map from images," MIT AI Memo 744, Cambridge, MA.

Ikeuchi, K., and Horn, B.K.P., [1981], "Numerical Shape from Shading and Occluding Boundaries," *Artificial Intelligence*, Vol. 17, p. 141-184.

Johansson, G., [1973], "Visual perception of biological motion and a model for its analysis," *Perception and Psychophysics*, (14), pp. 201-211.

Johnson, R.A., and Wichern, D.W., [1982], "Applied Multivariate Statistical Analysis," Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

Kender, J.R., [1980], "Shape from Texture,", Ph.D Thesis, Dept. Comp. Sci., Carnegie-Mellon University, Pittsburg, PA.

Kimia, B., and Zucker, S.W., [1983], "Deblurring gaussian blur," *IEEE Conf. Computer Vision and Image PROC.*, Washington, D.C.

Kitchen, L., and Rosenfeld, A., [1980], "Gray-level Corner Detection," TR-887, University of Maryland, College Park, MD.

Leclerc, Y.G., [1986], "The structure of discontinuities in one and two dimensions," Ph.D. Thesis (in preparation), Dept. Elect. Eng., McGill University, Montréal, Canada.

Leclerc, Y.G., and Zucker, S.W., [1984], "The Local Structure of Image Discontinuities in One Dimension", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (to appear).

Lee, C., and Rosenfeld, A., [1983]. "Improved Methods of Estimating Shape from Shading Using the Light Source Coordinate System." *University of Maryland TR-1277.*

Levine, M.D., O'Handley, D., and Yagi, G., [1973] "Computer Determination of Depth Maps." *Computer Graphics and Image Processing*, Vol. 2, No. 2, pp 131-151.

Link, N.K., and Zucker, S.W., [1985a]. "Sensitivity to Corners in Flow Patterns," Dept. Elect. Eng. TR-85-4, McGill University, Montréal, Canada.

Link, N.K., and Zucker, S.W., [1985b]. "Corner Detection in Curvilinear Dot Grouping." Dept. Elect. Eng. TR-85-5, McGill University, Montréal, Canada.

Mackworth, A.K., [1973]. "Interpreting Pictures of Polyhedral Scenes." *Artificial Intelligence*, Vol. 4, No. 2, pp 121-137.

Marr, D., [1976]. "Analysis of Occluding Contour," MIT AI Memo 372, Cambridge, MA.

Marr, D., and Poggio, T., [1977]. "A Theory of Human Stereo Vision," MIT AI Memo 451, Cambridge MA.

Marr, D., and Hildreth, E., [1980]. "Theory of Edge Detection," MIT AI Memo 518, Cambridge, MA.

Marr, D., [1982]. "Vision," W.H. Freeman & Co., San Francisco.

Marr, D., and Nishihara, H.K., [1977]. "Representation and Recognition of the Spatial Organization of Three Dimensional Shapes," MIT AI Memo 416, Cambridge, MA.

Marr, D., and Viana, L., [1980]. "Representation and Recognition of the Movewments of Shapes," MIT AI Memo 597, Cambridge, MA.

Nagel, H.H., [1978] "Analysis Techniques for Image Sequences," *PROC. IJCPR*, pp. 186-211.

Nagel, H.H., [1981], "On the Derivation of 3-D Rigid Point Configuration from Image Sequences", *PROC. IEEE Conf. Pattern Recognition and Image Processing*, pp. 103-108.

Nagel, H.H. and Neumann, B., [1981], "On 3D reconstruction from two perspective views," *PROC. IJCAI*, Vol. 2, pp. 661-663.

Nagel, H.H., [1983], "Displacement Vectors Derived from Second-Order Intensity Variations in Image Sequences," *Computer Vision, Graphics, and Image Processins*, Vol. 21, pp. 85-117.

Nevatia, R., [1976] "Depth Measurement by Motion Stereo," *Computer Vision, Graphics, and Image Processins*, Vol. 9, pp. 203-214.

Nevatia, R., and Binford, T.O., [1977], "Description and Recognition of Curved Objects," *Artificial Intelligence*, Vol. 8, No. 1, pp. 78-98.

Newman, W.M., and Sproull, R.F., [1979], "Principles of Interactive Computer Graphics," McGraw-Hill, New York.

Noble, P.B., & Levine, M.D., [1986], "Computer Assisted Analyses of Cell Locomotion and Chemotaxis," CRC Press, Boca Raton, Flordia, 1985.

O'Brien, N., and Jain, R., [1984], "Axial Motion Stereo," Technical Report CRL-TR-11-84, Dept. Comp. Sci., University of Michigan.

Pentland, A.P., [1982a], "The Visual Inference of Shape: Computation From Local Features," Ph.D Thesis, Department of Psychology, MIT, Cambridge, MA.

Pentland, A.P., [1982b], "Local Analysis of the Image: Limitations and Uses of Shading," *PROC. Workshop on Computer Vision Representation and Control*, August 23-25, Franklin Pierce College, Rindge, New Hampshire, pp. 153-161.

Pentland, A.P., [1985], "Perceptual Organization and the Representation of Natural Form," SRI TR 357, Menlo Park, CA.

Prazdny, K., [1979], "Motion and Structure from Opticl Flow," *PROC. IJCAI*, pp. 702-704.

Prazdny, K., [1980], "Egomotion and relative depthmap from optical flow," *Biological Cybernetics*, Vol. 36, pp. 87-102.

Requicha, A.A., [1980], "Representations for Rigid Solids: Theory, Methods, and Systems," *Computing Surveys*, Vol. 12, No. 4, pp. 437-464.

Richards, W., and Hoffman, D.D., [1984], "Codon Constraints on Closed 2D Shapes," MIT AI Memo 769, Cambridge MA.

Roberts, L.G., [1965], "Machine Perception of Three-Dimensional Solids," in *Optical and Electro-Optical Information Processing*, J.T. Tippett et al., (eds.), MIT Press, Cambridge, MA., pp 159-197.

Smith, G.B., [1979], "Using Enhanced Spherical Images for Object Representation.", MIT AI Memo 530, Cambridge, MA.

Smith, G.B., [1983a], "Shape from Shading: An Assessment," SRI Technical Note 287, Menlo Park, CA.

Smith, G.B., [1983b], "The Relationship Between Image Irradiance and Surface Orientation," *PROC. IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, D.C., June 19-23, pp 14-19.

Stevens, K.A., [1979], "Surface perception from local analysis of texture and contour," Ph.D Thesis, Dept. Elect. Eng. & Comp. Sci., MIT, Cambridge, MA.

Stevens, K.A., [1980], "Surface perception from Local Analysis of Texture and Contour," MIT AI Memo 512, Cambridge, MA.

Stevens, K.A., [1981], "The Visual Interpretation of Surface Contours," *Artificial Intelligence*, Vol. 17, p. 47.

Strang, G., [1980], "Linear Algebra and Its Applications," Academic Press Inc., New York.

Terzopoulos, D., [1982], "Multi-Level Representation of Visual Surfaces: Variational Principles and Finite Element Representations," in *Multiresolution Image Processing and Analysis*, A. Rosenfeld (ed.), Springer-Verlag, New York, 1983.

Terzopoulos, D., [1984], "Multiresolution algorithms in computational vision," in *Advances in Computational Vision*, S. Ullman (ed.), Ablex, new Jersey, to appear.

Tsai, R.Y., and Huang, T.S., [1981], "Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch," *IEEE Trans. Acouts., Speech, Signal Processing*, Vol. ASSP-29, pp. 1147-1152.

Tsai, R.Y., and Huang, T.S., [1984], "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 1, pp. 13-27.

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

Waltz, D.A., [1975], "Understanding Line Drawings of Scenes With Shadows," *The Psychology of Computer Vision*, P.H. Winston (ed.), McGraw-Hill, New York, pp. 19-80.

Waxman, A.M., and Ullman, S., [1983], "Surface Structure and 3-D Motion From Image Flow: A Kinematic Analysis," U. Maryland, Center for Automation Research, Technical Report CAR-TR-24, CS-TR-1332, College Park, MD.

Waxman, A.M., and Kwangyoen, W., [1984], "Contour Evolution, Neighbourhood Deformation, and Global Image Flow: Planar Surfaces in Motion," U. Maryland, Center for Automation Research, Technical Report CAR-TR-58, CS-TR-1394, College Park, MD.

Waxman, A.M., and Sarjavit, S.S., [1984], "Dynamic Stereo: Passive Ranging to Moving Objects From Relative Image Flows," U. Maryland, Center for Automation Research, Technical Report CAR-TR-74, CS-TR-1421, College Park, MD.

Webb, J.A., and Aggarwal, J.K., [1981], "Visual Interpretation of the Motion of Objects in Space," *PROC. IEEE Conf. Pattern Recognition and Image Processing*, Dallas, Texas, August 3-5, pp. 526-521.

Webb, J.A., and Aggarwal, J.K., [1982], "Shape and Correspondence," *PROC. Workshop on Computer Vision Representation and Control*, August 23-25, Franklin Pierce College, Rindge, New Hampshire, pp. 95-101.

Webster, [1953], "Webster's New World Dictionary of the American Language," Encyclopedic Edition, P.F. Collier & Son, New York.

Witkin, A.P., [1980], "Shape from Contour," Ph.D Thesis, MIT, Cambridge, MA.

Witkin, A.P., [1981], "Recovering Surface Shape and Orientation from Texture," *Artificial Intelligence*, Vol. 17, p. 17.

Witkin, A.P., [1983], "Scale-Space Filtering," *PROC. Eighth Int. Joint Conf. Artif. Intell.*, Karlsruhe, West Germany, pp. 1019-1022.

Witkin, A.P., and Tenenbaum, J.M., [1984], "On the role of structure in vision," in *Human and Machine Vision*, A. Rosenfeld and J. Beck (eds.), Academic Press, New York, pp 481-543.

Woodham, R., [1979], "Analyzed Curved Surfaces Using Reflectance Map Techniques," in

Artificial Intelligence: An MIT Perspective," Vol. 2, P.H. Winston and R.H. Brown (eds.), MIT Press, Cambridge, MA., pp 161-182.

Youssef, Y.M., [1982], "Quantification and Characterization of the Motion and Shape of a Moving Cell," Ph.D. Thesis, Dept. Elect. Eng., McGill University, Montréal Canada.

Zucker, S.W., [1975], "On the foundations of texture: A transformational approach," Ph.D thesis, Drexel University, Dept. Biomedical Eng., Philadelphia, PA.

Zucker, S.W., [1978], "Vertical and horizontal processes in low level vision," in *Computer Vision Systems*, E. Riseman and A. Hanson (eds.), Academic Press, New York.

Zucker, S.W., [1981], "Computer vision and human perception: An essay on the discovery of constraints," *PROC. Seventh Int. Joint Conf. on Artificial Intelligence*, Vancouver, B.C., Canada, pp. 1102-1116.

Zucker, S.W., [1982], "Early Orientation Selection and Grouping: Type I and Type II processes," Dept. Elect. Eng. TR-82-6, McGill University, Montréal, Canada.

Zucker, S.W., and Parent, P., [1982], "Multiple size operators and optimal curve finding," *PROC. International Conference on Pattern Recognition*, Germany, pp. 745-747.

Zucker, S.W., and Parent, P., [1984], "Curvature constraints from multiple-size asymmetric operators," *Multi-Resolution Image Processing*, A. Rosenfeld (ed.), Springer, New York.

Appendix A. Tangent Plane Interpolation

The tangent plane interpolation function is easily derived in terms of surface tilt and slant coordinates, τ and σ respectively. Choose a coordinate system centered at point x_o and let N_o be the normal to the surface at x_o. The equation of the tangent plane T_p at point x_o is given by

$$N \cdot (\vec{x} - \vec{x}_o) = 0, \tag{1}$$

where $\vec{x} \in T_p$.

We can define N_o alternately using tilt and slant coordinates defined as follows:

$$\sigma = \cos^{-1}(N_z)$$
$$\tau = \tan^{-1}(\frac{N_y}{N_z}) \tag{2}$$

where $N_o = \langle N_x, N_y, N_z \rangle$.

Rewriting (1) and rearranging to isolate z we obtain

$$xN_x + yN_y + zN_z = 0$$
$$z = \frac{xN_x + yN_y}{-N_z}. \tag{3}$$

The x, y, and z components of N_o are derived from (2) by simple manipulation

$$N_z = \cos(\sigma) \qquad \frac{N_y}{N_x} = \tan(\tau)$$
$$N_x = \cos(\tau)\sqrt{N_x^2 + N_y^2} \qquad N_y = \sin(\tau)\sqrt{N_x^2 + N_y^2}. \tag{4}$$

Now if N_o is the unit normal vector at x_o (n.b. this is precisely what we obtain from the shape from shading computation), then

$$\sqrt{N_x^2 + N_y^2 + N_z^2} = 1$$

$$\sqrt{N_x^2 + N_y^2} = \sqrt{1 - N_z^2}. \tag{5}$$

Substituting the expression for N_z from (4) we obtain

$$\sqrt{N_x^2 + N_y^2} = \sin(\sigma),$$

i.e.

$$N_y = \sin(\tau)\sin(\sigma) \qquad N_z = \cos(\tau)\sin(\sigma). \tag{5}$$

Substituting the components of N_o into (3) results in

$$z = \frac{x\sin(\sigma)\cos(\tau) + y\sin(\sigma)\sin(\tau)}{-\cos(\sigma)} \tag{6}$$

Thus, relative to a point x_o

$$\Delta z = \frac{\Delta x\sin(\sigma)\cos(\tau) + \Delta y\sin(\sigma)\sin(\tau)}{-\cos(\sigma)} \tag{7}$$

Appendix B. Recursive Integration Algorithm

The depth map corresponding to a surface S is obtained by applying the following algorithm to the unit normal vector field spanning S. It assumes that this field has been decomposed into piecewise continuous regions and that the depth of at least one point on S is known[47].

S1: Push the coordinates and depth of each known fixed point onto a stack K.

S2: Enter the depth of each fixed point at the appropriate coordinates of the depth map corresponding to S. The depth at all other coordinates is initialized to zero.

S3: Randomly select and remove a fixed point from the stack. Re-order K so as to fill the hole generated by the removal.

S4: Interpolate the depth of the fixed point along its tangent plane (Appendix A) to each of its near-neighbours. If the near-neighbour is part of the same region as the fixed point, push its coordinates and interpolated depth onto the stack K. Fill in the depth map with each interpolated depth value.

S5: Repeat steps S3 and S4 until the stack is empty.

Discussion

Upon completion, the depth map will contain depth values for each point on S that can be reached by traversing the surface from a known fixed point. Obviously, boundary conditions (i.e. fixed points) must be provided for each region on S that is isolated by virtue of discontinuities in depth. When boundary conditions are not available, an attempt may be made to estimate the depth at a boundary on the basis of the depth at a neighbouring region, but not in all circumstances. Discontinuities in depth correspond to self-occlusions of the surface, occlusions from nearby surfaces, or creases on the surface. There is no way

[47] If this condition is not met, an arbitrary point is chosen as reference and initialized to a depth of zero.

to resolve depth changes due to occlusion, however creases can be sometimes be handled by interpolation. The key is being able to classify the nature of the discontinuity. Surface slant, σ is a useful indicator in this regard as it is known to roll off to 90° at occluding contours.

The path of integration taken by the algorithm resembles that of a brush fire where the flames fan out from the center in all directions at approximately the same speed. This is effected by randomly selecting previously computed depth points from the stack, and also distributes the accumulated error evenly over the surface. To minimize the error even further, the procedure is repeated for several iterations and the depth maps for each iteration averaged. For each iteration the path of integration from the reference to a particular point on the surface is different. Thus, by taking the average over several iterations, the accumulated error at a point is averaged over a larger part of the surface. The procedure is allowed to iterate until the maximum depth change at any point on the surface falls below a specified threshold. In experiments with normal fields computed from shaded images, this algorithm has demonstrated rapid convergence, and usually within 10 iterations.

Appendix C. Tables 1-5

Sphere Generation Parameters

Exact L: $< 0.4160, -0.2774, 0.8660 >$
Estimated L: $< 0.4849, -0.3332, 0.8086 >$ Δ°: 6.05°
Forced L: $< 0.4849, -0.3332, 0.7500 >$ Δ°: 19.35°

A. Using actual L

	Estimated	Actual	Δ%
Centroid X	0.0	1.5900	1.24%
Centroid Y	0.0	1.7100	1.34%
Centroid Z	0.0	2.0500	1.60%
Axis A	0.9881	1.0	1.19%
Axis B	1.0170	1.0	1.70%
Axis C	0.9587	1.0	4.13%

B. Using estimated L

	Estimated	Actual	Δ%
Centroid X	0.0	1.3700	1.07%
Centroid Y	0.0	1.3500	1.05%
Centroid Z	0.0	0.3840	0.30%
Axis A	0.9849	1.0	1.51%
Axis B	1.0064	1.0	0.64%
Axis C	1.0259	1.0	2.59%

C. Using forced L

	Estimated	Actual	Δ%
Centroid X	0.0	-0.5000	0.39
Centroid Y	0.0	1.7300	1.35
Centroid Z	0.0	0.0300	0.02
Axis A	1.0484	1.0	4.84%
Axis B	1.0367	1.0	3.67%
Axis C	0.9843	1.0	1.57%

Table 1

Ellipsoid Generation Parameters

Exact L: $< 0.1504, -0.0868, 0.9848 >$
Estimated L: $< 0.1555, -0.1432, 0.9774 >$ $\Delta°$: 3.27°
Forced L: $< 0.1555, -0.1432, 0.9000 >$ $\Delta°$: 22.76°

A. Using actual L

	Estimated	Actual	Δ%
Centroid X	-0.5400	0.0	0.42%
Centroid Y	0.5740	0.0	0.45%
Centroid Z	0.8925	0.0	0.70%
Rotation X	-4.0354	0.0	4.48%
Rotation Y	2.4400	0.0	2.71%
Rotation Z	0.2225	0.0	0.25%
Axis A	0.9836	1.0000	1.64%
Axis B	0.7176	0.7000	2.51%
Axis C	0.3554	0.3500	1.54%

B. Using estimated L

	Estimated	Actual	Δ%
Centroid X	2.2278	0.0	1.74%
Centroid Y	1.3689	0.0	1.07%
Centroid Z	1.0101	0.0	0.79%
Rotation X	2.9441	0.0	3.27%
Rotation Y	2.1095	0.0	2.34%
Rotation Z	1.8816	0.0	2.09%
Axis A	0.9597	1.0000	4.03%
Axis B	0.7265	0.7000	3.79%
Axis C	0.3613	0.3500	3.23%

C. Using forced L

	Estimated	Actual	Δ%
Centroid X	-1.1633	0.0	0.91%
Centroid Y	1.6122	0.0	1.26%
Centroid Z	1.0082	0.0	0.79%
Rotation X	2.9224	0.0	3.25%
Rotation Y	2.2325	0.0	2.48%
Rotation Z	0.9676	0.0	1.08%
Axis A	0.9719	1.0000	2.81%
Axis B	0.7052	0.7000	0.52%
Axis C	0.3618	0.3500	3.37%

Table 2

Elongated Ellipsoid Generation Parameters

Exact L: $< 0.0000, 0.0000, 1.0000 >$
Estimated L: $< 0.0000, -0.1741, 0.9847 >$ $\Delta°$: 10.04°
Forced L: $< 0.1555, -0.1432, 0.9000 >$ $\Delta°$: 22.76°

A. Using actual L

	Estimated	Actual	Δ%
Centroid X	0.8004	0.0	0.63%
Centroid Y	0.5382	0.0	0.42%
Centroid Z	1.3488	0.0	1.05%
Rotation X	-0.0616	0.0	0.07%
Rotation Y	1.6553	0.0	1.84%
Rotation Z	0.1134	0.0	0.13%
Axis A	1.1863	1.0000	18.63%
Axis B	0.1920	0.2000	4.00%
Axis C	0.5274	0.5000	5.48%

B. Using estimated L

	Estimated	Actual	Δ%
Centroid X	0.9642	0.0	0.75%
Centroid Y	1.0568	0.0	0.83%
Centroid Z	1.0968	0.0	0.86%
Rotation X	0.3112	0.0	0.35%
Rotation Y	0.4500	0.0	0.50%
Rotation Z	-0.0315	0.0	0.04%
Axis A	1.1281	1.0000	12.81%
Axis B	0.1851	0.2000	7.45%
Axis C	0.5404	0.5000	8.08%

C. Using forced L

	Estimated	Actual	Δ%
Centroid X	0.4768	0.0	0.37%
Centroid Y	1.0253	0.0	0.80%
Centroid Z	1.4662	0.0	1.15%
Rotation X	-0.2102	0.0	0.23%
Rotation Y	6.1904	0.0	6.88%
Rotation Z	-0.3948	0.0	0.44%
Axis A	0.9533	1.0000	4.67%
Axis B	0.1870	0.2000	6.50%
Axis C	0.5773	0.5000	15.46%

Table 3

Torus Generation Parameters

Exact L:	$< 0.0000, 0.0000, 1.0000 >$	
Estimated L:	$< 0.0163, -0.0716, 0.9974 >$	Δ°: 4.13°
Forced L:	$< 0.0163, -0.0716, 0.9000 >$	Δ°: 20.00°

A. Using estimated L

	Estimated	Actual	Δ%
X Section Ratio:	1.1015	1.0	10.15%

B. Using forced L

X Section Ratio:	1.1174	1.0	11.17%

Table 4

Full vs. Partial Surface Estimation

A. Using full surfaces

	Estimated	Actual	Δ%
Centroid X	-0.5400	0.0	0.42%
Centroid Y	0.5740	0.0	0.45%
Centroid Z	0.8925	0.0	0.70%
Rotation X	-4.0354	0.0	4.48%
Rotation Y	2.4400	0.0	2.71%
Rotation Z	0.2225	0.0	0.25%
Axis A	0.9836	1.0000	1.64%
Axis B	0.7176	0.7000	2.51%
Axis C	0.3554	0.3500	1.54%

B. Using partial surfaces

	Estimated	Actual	Δ%
Centroid X	1.0553	0.0	0.83%
Centroid Y	0.7477	0.0	0.58%
Centroid Z	10.3020	0.0	8.05%
Rotation X	6.9043	0.0	7.67%
Rotation Y	0.0436	0.0	0.05%
Rotation Z	1.6372	0.0	1.82%
Axis A	0.5872	1.0000	41.28%
Axis B	0.5068	0.7000	27.60%
Axis C	0.1553	0.3500	55.62%

Table 5

Appendix D. Classification of Quadratic Surfaces

From Hall et al. [1982] p. 51.

Classification of quadric surfaces.

		Proper quadrics $A \neq 0$			**Improper (degenerate) quadrics** $A = 0$				
		$A > 0$		$A < 0$	**Cones and cylinders** $A' \neq 0$		**Pairs of planes** (degenerate quadrics) $A' = 0$		
		$A'I$ and J both > 0	$A'I$ and J not both > 0		$A'I$ and J both > 0	$A'I$ and J not both > 0	$A'' > 0$	$A'' < 0$	$A'' = 0$ $A''' \neq 0$
Central quadrics $D \neq 0$	DI and J both > 0	**No real focus** (imaginary ellipsoid)	...	**Real ellipsoid**	**Point** (in finite portion of space; vertex of imaginary elliptic cone; point ellipsoid)	...	...	...	...
	DI and J not both > 0	...	**Hyperboloid of one sheet**	**Hyperboloid of two sheets**	...	**Elliptic cone** (degenerate hyperboloid)	...	...	...
Noncentral quadrics $D = 0$	$J > 0$	...	...	**Elliptic paraboloid** (of revolution if $I^2 = 4D$	**No real locus** (imaginary elliptic cylinder)	**Real elliptic cylinder** (circular if $I^2 = 4D$	**Straight line** (degenerate elliptic cylinder; intersection of imaginary planes)	...	...
	$J < 0$	...	**Hyperbolic paraboloid**	...	...	**Hyperbolic cylinder**	...	**Two real planes** intersecting in finite portion of space (degenerate hyperbolic cylinder)	...
	$J = 0$ $I \neq 0$	...	...	...	...	**Parabolic cylinder**	**No real locus** (imaginary parallel planes)	**Two real parallel planes** (degenerate parabolic cylinder)	**One real plane** (coincident parallel planes)
Rank of the 4th-order square matrix $\mathbf{A}_{ik}$		4			3		2		1