ACTIVE 3D SURFACE MODELING USING PERCEPTION-BASED, DIFFERENTIAL-GEOMETRIC PRIMITIVES

by

Liangyin Yu

A dissertation submitted in partial fulfillment of the

requirement for the degrees of

Doctor of Philosophy

(Computer Science)

at the

UNIVERSITY OF WISCONSIN – MADISON

1999

Abstract

Computational vision is about why a biological vision system functions as it does and how to emulate its performance on computers. The central topics of this thesis are how a differential geometry language can be used to describe the essential elements of visual perception in both 2D and 3D domains, and how the components of this geometric language can be computed in ways closely related to how the human visual system performs similar functions.

The thesis starts by showing that at the earliest stage of vision, biological systems implement a mechanism that is computationally equivalent to computing local geometric invariants at the two-dimensional curve level. The availability of this information establishes the foundation for computing components of a differential geometry language from sensory inputs. The mathematical framework of scale space that makes this computational approach possible, likewise, has its biological basis.

On the other hand, visual perception is a global phenomenon that occurs generally in a 3D space. To understand this process and design computational systems that have comparable performance to humans requires specification of how a 2D local computational mechanism can be used in this global 3D environment. This goal is achieved through two steps. First, a global surface representation formulation is extended from the 2D framework. It is shown how local geometric features that are sparse and perceptually meaningful can be naturally used to represent global 3D surfaces. Second, active motion by an observer is introduced as an additional dimension to the data set so that the observer becomes mobile and can react to observations or verify hypotheses actively. This also makes dynamical data such as optical flow available

to the observer. These added abilities enable the observer to perform tasks such as surface recovery and 3D navigation. In addition, the modeling process of 3D objects is naturally constrained by the computational resources available to the observer so that the model is inherently incremental.

This thesis contributes in the following areas: (1) direct computation of 2D differential geometric invariants from images using methods comparable to the human vision system, (2) perception-based global representations of 2D and 3D objects using geometric invariants, (3) novel methods for optical flow computation and segmentation, and (4) active methods for global surface recovery and navigation using both stationary contours, apparent contours and textured surfaces.

Acknowledgments

My immense thanks and appreciation go to Chuck Dyer, my insightful advisor. His input and advice on the research content and effective communications are invaluable. I cannot think of a better advisor to work with.

Special thanks go to Professors Nicola Ferrier, Damon Tull, Jude Shavlik and Michael Gleicher for taking time to read this work as members of the thesis committee. It was also beneficial discussing with my former fellow student, Steve Seitz, along the years.

I would also like to express my appreciation and love to my family who support me unconditionally along the years of my study. Finally, I would like to dedicate this thesis to my mother, who did not have a chance to see how a great story comes to its final chapter. I am forever in debt to her love and tenderness.

Contents

List of Figures

Chapter 1

Introduction

Computational vision is about both visual perception and how computers can perform comparable tasks as a biological system. So, we study computational vision to understand the mechanism underlying the human visual system, and to design systems that are comparable to the performance of a biological system. In this research, *visual perception* is used in the context of the human visual system, while *computation* is applied to both biological systems and computers.

Research on computational vision, inspired by the incredible visual abilities in the biological world, has a long history of establishing theories that are based on signal processing and geometric modeling. Computationally, both approaches use descriptions that require large amounts of information, either in the form of raw data (e.g., pixels) or analytical descriptions (e.g., spline parameterization). The adoption of these approaches is a direct consequence of failing to constrain visual tasks effectively and, as a consequence, inordinate amounts of computational resources have to be spent in order to derive general descriptions that are useful for all tasks. This is particularly true when vision is treated to some extent as an inverse problem of computer graphics, i.e., designing algorithms to transform image data to geometric languages used in computer graphics (Figure 1.1).

One critical issue at the earliest stage of vision is to design highly data-selective processing

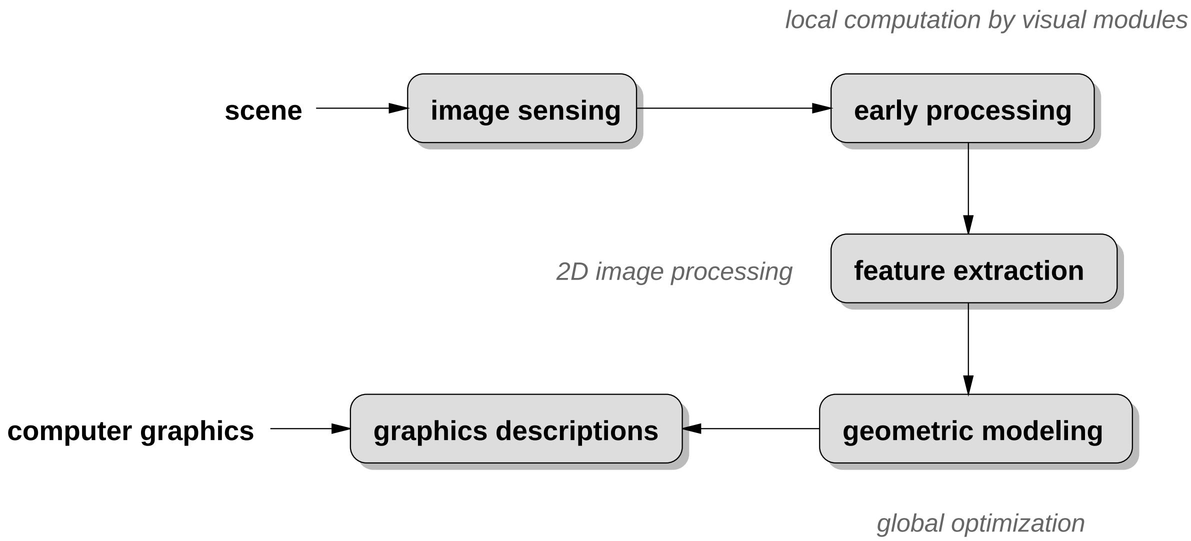


Figure 1.1: General approach to computational vision.

modules that extract those relevant data that will be useful for all tasks. This is the concept underlying the design of “early-processing” modules that are tuned to critical visual features. The study of existing biological mechanisms provides insight into how the capability of a vision system is specified as well as the nature of the computation performed. The results from this study also enable us to describe formally the relationship between computational models and visual perception.

In spite of this insight and sound mathematical modeling techniques, it becomes increasingly difficult when it comes to the interpretations of the computation and specifications of tasks. We will need certain “contexts” to interpret the results of computation and evaluate how well a task is performed. In this research, the major task is 3D object representation and recognition, and the context is both perceptual and geometric.

The research presented in this thesis suggests that the information necessary for completing a visual task can be highly constrained and put in effect early in the processing pipeline. With an adequate choice of the underlying geometric language, the results available from the computation at the early stage can be interpreted in both perceptual and geometric contexts. These interpretations lead us gradually to increasingly more abstract and more powerful concepts such as curves and surfaces in the geometric language. However, in contrast to methods in computer graphics, these geometric concepts have direct computational and perceptual components, which allow us to compute, for example, surfaces based on the input sequence of images and their local properties computed at the visual system front-end. In addition, the perceptual interpretations enable the observer to actively seek out the missing parts of the computation or verify current hypotheses regarding object shape.

1.1 Problems and Core Issues

The initial computational problem encountered in vision is to determine which parts of the image data should be retained for further processing. This selection process is imperative in view of the overwhelming data available at the photoreceptors, when considering the dynamic nature of vision in both the spatial and temporal domains. On the other hand, what is discarded as irrelevant cannot be restored later. Consequently, this data selection process also determines the nature of the vision system, including what tasks it can perform and how much resource is needed in order to perform an individual task. Hence, the decision can only be reached by considering the tasks to be performed. The diversity of vision systems in nature demonstrates how resources can be used differently and effectively for the design of a system when facing different tasks. In computational vision, the main focus is those highly sophisticated systems such as a human's with object representation and recognition capabilities. Hence the first problem to be investigated is:

Problem 1: What are the essential computations at the front-end of a vision system that are capable of computing representations for complex shapes? — The core issue here is how the system can retain sufficient information to complete the essential tasks within the constraint of available computational resources. The computational models at the front-end for the task of shape representation are studied in Chapter 2.

Since our major focus is oriented toward the task of object representation and recognition, a language is needed to describe the task formally and to associate the language with the data through computational procedures. This association closely relates to what is considered to be "features" of an object and how the object can be represented by these features. When the language is chosen to be geometric, we need to solve the following problem:

Problem 2: What kind of geometric language can be used to describe relevant perceptual results in human vision and how can the elements of the language be computed from the data received at the front-end? — The core issue here is to establish a formal geometric description for the task and specify how the elements of the description can be derived from the data. The geometric language and its components are described in Chapter 3. The specific computations of these components from images are presented in Chapter 4.

When the geometric language is chosen to be analytic or has components that are locally smooth (e.g., differential components), it has a natural relationship to the functionality of the system front-end, since the front-end is characterized by local computations. However, visual perception is a macroscopic or global activity and we have to answer the question:

Problem 3: How are the global properties of perception related to the results of local computation as dictated by our choice of geometric language? — The core issue here is to explicitly specify the global properties and show how the computation from local to global domain can be achieved. This problem is studied in Chapter 5 and Chapter 8 by employing the global curve model in Chapter 3 and the global surface model in Chapter 8.

The components of the geometric language are what is to be computed from the raw data. For an analytic geometric formulation, the computation of its components requires infinite time to complete because of the infinite resolution implied. When it is necessary to terminate the computation due to finite computational resources, the resulting representation must have a well-defined relationship to other representations acquired from different resource requirements. Hence the following problem arises:

Problem 4: When should the computation of the language terminate and what is the relationship between the representations computed under different resource constraints? — The core issue here is to represent the results of the computation when it terminates and to analyze the relationship between different representations such that there is a smooth transition across different computations (incremental modeling). Incremental modeling for 3D representations of surfaces are presented in Chapters 8.

The information used for object representation is part of the sensory input and all the above problems will be studied under the assumption that those input data are useful for the task. This is the passive view of how the task is handled. In general, the assumption can be false and we should not impose, as is commonly done in computer vision, further assumptions that agents outside the system can supply the relevant data. To close the loop for data acquisition, we should also answer the following question:

Problem 5: How can an autonomous system actively seek out information based on what has already been observed? — The core issue in this problem is to specify sound procedures that can be effectively used to navigate and acquire essential information that are missing for completing a task. This problem is handled within the active vision paradigm. The case of computing surfaces from stationary surface curvilinear features is studied in Chapter 6, and active 3D navigation is studied in Chapter 8, while the computation of optical flow is studied in Chapter 7.

1.2 Thesis Statement and Contributions

This research covers two major categories of problems in computer vision: the computation of geometric features from 2D images, and 3D object modeling by an active observer. The results are mathematical formulations of how perceptually meaningful, 2D geometric properties can be embedded in early processing modules, and how active processes can be used to facilitate 3D modeling and object recognition based on the results of the 2D computation.

The central thesis of this research is:

1. *The language employed by visual perception to represent objects is intrinsically both perceptual and geometric, and this nature has to be reflected in all stages of information processing*.

2. *The nature of visual information processing is active rather than passive*.

The major contributions of this research can be categorized as follows (Figure 1.2).

1.2.1 Contributions

Geometric Feature Computation Using Receptive Fields The mathematical modeling of visual information processing is formulated based on foundations from the biological nature of the human visual system. This modeling is effective in bridging the gap between visual perception and the geometric language. By employing the framework of scale space, the information domain is effectively smooth for differential computations and, yet, can be interpreted globally for perception. It is shown how intrinsic properties of 2D curves can be locally computed from an image using filters from a receptive field family.

Global Curve Modeling and Surfaces from Stationary Contours The computed 2D curve properties are used to construct global object models used by a new scheme for 2D object recognition. The 2D framework and results are readily extended to 3D when we consider stationary curvilinear markings on a surface. It is shown that an active observer can navigate the surface to recover the surface geometry around these markings.

Optical Flow Computation and Segmentation For a textured surface, new methods for optical flow computation and segmentation are developed so that the observer can determine the object boundaries using local motion. We also demonstrate how an active observer can control its movement in order to control how an optical flow field is decomposed. This ability allows the observer to choose vantage points to obtain better spatial relations with the surface during navigation.

3D Navigation and Perception-Based Incremental Modeling For the problem of 3D surface representation, a new theory of 3D incremental modeling is presented so that an observer can infer feature points and curves on a surface that are both intrinsic to the surface geometrically and meaningful for visual perception. We also present navigation theorems that enable the observer to actively verify hypotheses formed from the current observation. It is proved that, computationally, navigation will terminate for the task of surface recovery.

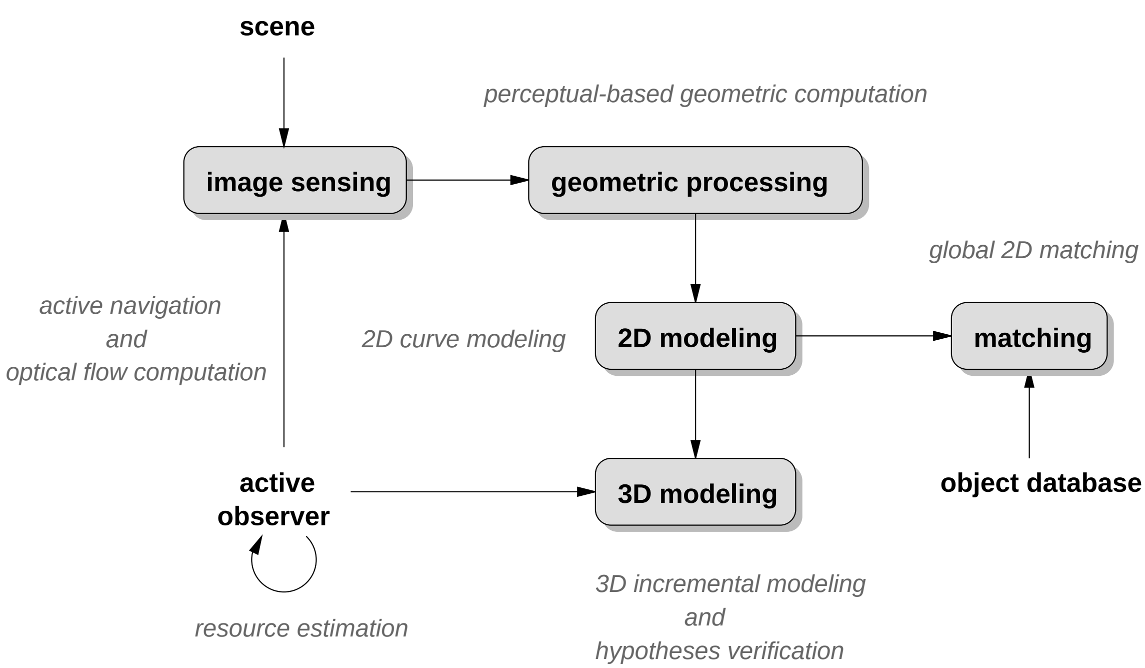


Figure 1.2: Approach and contributions.

1.3 Methods and Assumptions

1.3.1 Base Filters

The computational models at the system front-end are based on filters of "limited bandwidth in both space and frequency domains." These filters are chosen on the basis of the assumption that there are both spatial and spectral components in visual information processing. Consequently, scale-space variants of Gabor and Gaussian filters are natural candidates for the modeling of these components. This choice is also supported by results from physiological research.

1.3.2 From Local to Global Domain

Visual perception is a global activity [8], even though all the computations at the system front-end are local. Hence, it is assumed that a computational theory of vision needs to explain how this gap between local computation and global perception can be bridged.

1.3.3 Geometric Language, Perception and Representation

The assumption is made that it is the geometric events occurring in 3D Euclidean space that are perceived and should be computed from the raw image. In this research, the geometric language is chosen to be differential in nature, i.e., the measure of distances between data properties is infinitely smooth. This idealization is appropriate by making scale-space a foundation of the modeling. A consequence of this approach is the emergence of "families" of base filters (i.e., the Gaussian functions) that are indexed by the scale-space parameters. On top of the language, it is also assumed that the representation is shape-oriented so that there are well-defined processes that can recover the object shape from the representation.

1.3.4 Task-Oriented, Closed-Loop Approach and Active Vision Paradigm

The logical structure of a vision system has two parts: (1) to compute high-level geometric properties from low-level data, and (2) to ensure the coherence of the computation by verifying the high-level results in the low-level domain. This ability can be acquired if we adopt a "closed-loop" processing model and provide the observer with voluntary mobility. That is, methods of active vision will be used to establish the hierarchy of geometric structure from low-level data to high-level object representation. This is achieved by actively seeking out relevant information and verifying hypotheses in case of insufficient information. Furthermore, it also implies that the observer can incorporate the task specification so that only task-oriented information needs to be located.

During the limited span of both spatial and temporal intervals, the imaging information can be processed by a system without closing the processing loop (i.e., there is no feedback from later stages to earlier stages). Since the propagation and processing of signals require time, it is always possible to define such a time span so that only open-loop systems need to be considered. However, this restriction greatly limits the tasks that can be performed by the system (e.g., only the reflexive movements that bypass the cerebral cortex). It is hypothesized that only the early stage of the system can benefit from a complete open-loop design.

1.4 Motivation and Related Work

1.4.1 Visual Information Model

There are two extremes in the spectrum of human-computer vision problems: the raw data (signal) available at the sensory front-end and the 3D modeling of the physical world. The analytical capability of methods from signal processing simplifies greatly the problem formulation in computational vision, but applying these methods directly in vision makes it difficult to interpret the results in terms of visual perception. It is also inherent in these methods that the information is uniform across both space and time, while visual perception is more than discriminating signal and noise. Without an information model to interpret which part of the signal is relevant, computational problems in vision will be very difficult to solve (Figure 1.3).

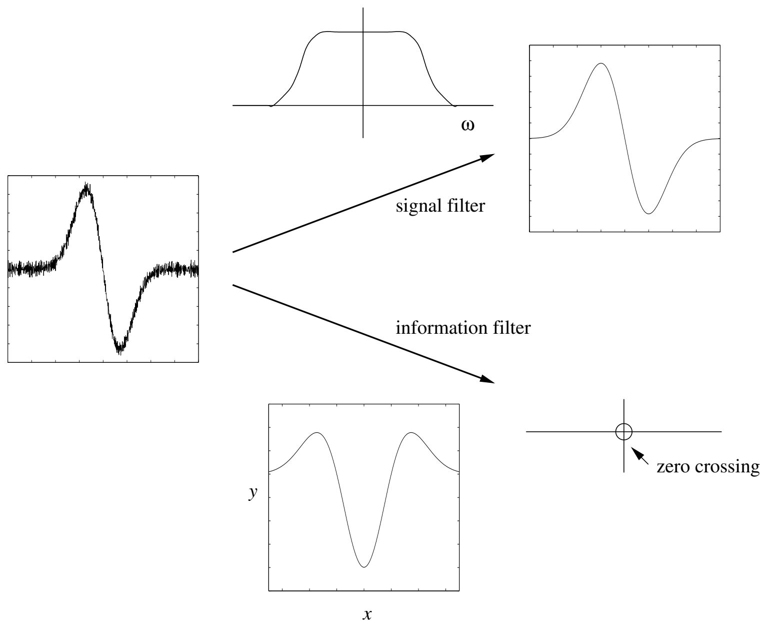


Figure 1.3: The signal model only concerns the separation of noise from signal, while the information model specifies that only zero-crossings are relevant.

Perception as Part of the Model The emergence of Marr's *primal sketch* representation [77] drew results from physiology and psychophysics and made explicit statements about what is computed for shape representation. This model motivated a number of theories regarding early processes of vision and culminated in theories such as optimal edge detection [22], 2D interpretation of 3D shapes [11], and integration of visual modules [4]. One of the consequences of this model is the critical decision of which part of the input data is relevant for intended tasks at the early stages, and this decision is based on visual perception. The computational problem regarding an appropriate (geometric) information model for shape representation and its relationship to visual perception is one of the major themes in this thesis.

Information models such as the primal sketch are 2D-based, and the inverse problems of inferring 3D information from these 2D models are ill-posed [13]. Consequently, additional assumptions have to be made in order to make this process feasible. For example, by imposing assumptions regarding the nature of image formation, some properties of the object surface can be inferred from the primal sketch (shape-from-X approaches [51]). Or, by minimizing certain error energy functions to regularize the solution space [86] or integrate visual cues for surface representation [99].

Hypothesis Verification as Part of the Model Imposing hypotheses is a way of introducing additional structure on the relatively primitive data and transforming it into more structured information. However, the transformations resulted from these hypotheses are not intrinsic to visual perception and hence is superficial to the problem. For a biological system, the route from 2D information model to 3D shape representation is naturally an active one, which involves exploring and navigating the 3D environment. There is no additional assumption being imposed here. Instead, assumptions are verified or falsified by actively collecting evidence. This process is another major theme of this research.

1.4.2 Early Vision and Biological-Based Modeling

Early stages of visual information processing are characterized by local computation that uniformly selects the parts of the raw signal that are useful for all tasks. The implementation of the early modules for these stages also define the capability of the vision system per se. These computations are generally considered to be the first stage of a hierarchical structure of a vision system. As such, research efforts have been made to "optimize" these processes so that subsequent stages can rely on these initial computations. Other than properties that are directly related to local computation such as edge detection and optical flow, it is also considered essential to derive increasingly global properties (e.g., surface geometry and motion segmentation) from these stages. However, devising local operators for these goals has proven exceedingly difficult, even though there is evidence that biological systems are able to obtain global properties from local computations [53, 73].

Static and Dynamic Modeling According to their intended inputs, the models of the early visual modules generally are of two categories: static and dynamic, though the physiological foundation of both is essentially dynamic (e.g., a stabilized image fades quickly without saccadic movement [75]). The static models are developed for the processing of static images, and are proved to be effective for operations such as edge detection and texture analysis [77]. The dynamical models, on the other hand, are for motion analysis, in which the input images are also sampled in the temporal domain.

Receptive Fields The biological mechanisms responsible for the local computations in early vision are the "receptive fields" [3, 30, 52, 71], whose computational properties are modeled extensively in vision research [27, 66, 91]. Both spatial and frequency domain methods have been used for modeling early vision modules [27, 28, 29, 92].

Scale Space One of the parameters inherent in the receptive field modeling is the size of each field. In terms of the sampling topology implemented by the receptive fields, the size parameter defines a *scale space* [54, 58, 114, 116] for what is being computed. This formulation is essential in bridging the local computation [37, 65] and global representation gap. Since it is embedded in the receptive field modeling, the information structure computed by the receptive field naturally inherits a scale space structure as well [37, 67].

1.4.3 2D Geometric Modeling

Image contours exhibit good correspondence between image structure and the physical world at early stages of visual perception. They also bridge the gap between local computations of the 2D image signal and perception (organized globally). When considered along with 3D geometry, 2D contours can even be used to constrain 3D surfaces to a certain extent [11, 88]. For a given 2D curve, there is no single description that is canonical in its geometric content. However, there are descriptions with parameters such as curvature extrema that are meaningful perceptually [8, 35, 36, 87], and invariant under restricted affine transformations (rigid transformations plus uniform scaling) [49]. In other words, these parameters characterize the essential geometric structure of an image contour. This observation, combined with the acknowledgment that the structure of an image contour can be described at multiple scales, has led to such concepts as curvature space [87, 81], scale-space primal sketch [72], curvature primal sketch [7], and curvature scales [67].

Computation of Contours In order to make the structure of an image contour explicit, a geometric model [67, 80] of the contour needs to be computed from the image, which is also implemented by biological systems [32, 61]. In computational vision, two methods are commonly employed: (1) local edge detection followed by global curve tracing [81, 84], and (2) global interpolation or energy optimization [57, 118]. The difficulty with the first approach is the strictly one-dimensional sequential processing model and data dependency, e.g., the estimation of curvature depends exclusively on the current edge locations and their estimated tangents, which, in turn, depend only on the resolution provided by an edge detector. Hence errors produced in early stages propagate to and are amplified by all later stages. Recognition of this fact results in the general consensus that higher order geometric invariants of image contours (such as curvature) are noisy and unstable computationally [112]. The method of energy optimization, on the other hand, requires a careful design of energy terms to stabilize the results, and both methods do not perform well across tangent or curvature discontinuities.

1.4.4 2D Object Recognition

The problem of 2D object recognition can be defined as: Given a 2D object described by a representation method and a set of known instances described by the same method, identify efficiently the degree of similarity between the object and the known instances, and, if necessary, a set of well-defined transformations that are needed in order to make the similarity explicit. The method of representation should be stable under noise and should preserve 2D shape information. The method of identification should not be dependent on the number of known instances in terms of computation time.

One of the major issues in object recognition is to find object representations and matching processes that are invariant under view variations and are computationally efficient. Invariance with respect to a specific kind of variation requires representations that either are independent of the variation or have a well-defined behavior in the presence of the variation. The matching process identifies similarity between object representations and a set of known instances (i.e., models), and should be as independent as possible of the size of the database. In addition, any proposed representation and matching method must be stable in the sense that limited variations in the input produce limited variations in the representation and the performance of the matching process. This stability enables us to define a metric for matching as well as dealing with some of the invariance issues mentioned above. Commonly used invariants are *projective* and *affine* invariants as well as proper subsets of these two groups, e.g., scaling and rigid transformations.

2D Representation For 2D object representations, the descriptors or primitives are curvilinear contours [114], regions or image templates [55, 111]. Object representations by curves are associated with perceptual organization [115] and curve partitioning [35, 36], in which perception-related criteria are used to partition the curves of object boundaries. Both spatial and Fourier domain representation methods have been developed. Spatial domain methods generally isolate interesting parts (e.g., curvilinear features) from background noise and establish one or more descriptions based on these isolated parts [114], while Fourier methods are commonly used for image templates.

To address the problem of view invariance in 2D object recognition, the 2D primitives used for the representation are either constructed from invariants for a carefully selected set of features (primitive or invariance-based)[6, 82] or from multiple views (appearance or view-based) [63]. Invariants can be selected either from local or global representations. In general, local representations (differential-based) of curves are not stable against noise but do preserve curve information, while global representations (e.g., moment-based [98]) are more stable but do not preserve curve information. In cases where image templates are used directly as known instances, either partial transformation (deformations) functions (scaling and translation) are recovered during a point-based matching process [55], or an invariant transformation such as the magnitude of the Fourier transform is used [111].

2D Matching The matching process generally involves graph-matching, relaxation methods or topological distance measures. In cases where searching is involved in the matching process, the computational complexity with respect to the number of known instances in the database has been a difficult problem and has been tackled from the perspectives of both searching strategies [41] and invariance indexing [89]. Techniques such as hashing and indexing have also been developed in order to solve this problem [12, 89, 95, 103]. A major problem with indexing is the many-to-one mapping from object representation to indexed space, i.e., the representation can not be recovered from the index itself. Under this condition, the probability that multiple objects or noise resolve to the same index is high. Consequently, algorithms with time complexity that is approximately independent of the size of the object database are needed.

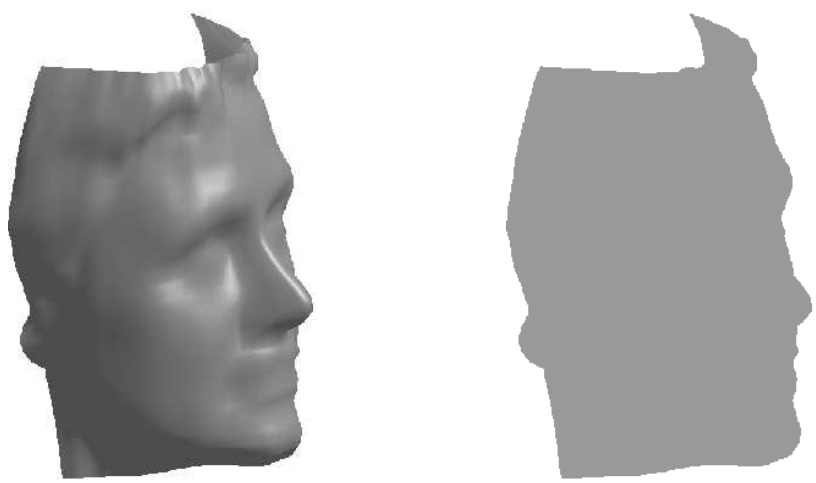
1.4.5 3D Representations

3D computer vision, in which 3D geometry is inferred from image analysis, has generally been considered as a complementary problem of computer graphics where images are produced by analyzing the interaction of light and 3D geometry. In either problem domain, 3D geometry is considered a substrate and essential for problem solving. However, the geometric models in computer graphics are designed to generate visually appealing images, while in computer vision, local and usually differential, properties are identified first and assembled later into geometric models by applying additional global constraints. The resulting surface shapes in vision are often represented in a way closely related to computer graphics such as quadrics [34], superquadrics [9, 85], and dynamic meshes [99]. Consequently, the criteria used in the representation are generally without perceptual significance. This causes difficulties in determining how the results from the front-end should be connected to the 3D representation. In other words, the choice of geometric models dictates what should be computed and, often, how to compute it. In contrast, the "reconstruction" of a surface from its representation as a means of, for example, visual communication can naturally employ computer graphics techniques, since the process need not go through the same hierarchy of the visual processing to fulfill the purpose. A complete framework for 3D representation should, however, contain both parts, i.e., representation and reconstruction.

Representation and Reconstruction of Surfaces Other than the desirable connection between a representation and visual perception, viewpoint or even affine invariance is needed for tasks such as 3D object recognition. This subject was studied extensively in the form of geometric invariance [82]. For a smooth region of a surface, intrinsic surface properties of differential geometry are usually employed [31]. Gaussian curvatures, critical points [14, 90], and extended Gaussian images [50] are among the commonly used intrinsic surface properties. To reconstruct surfaces from a point-based, local representation, both variational methods such as thin plate [41] and energy minimization [16, 18, 99] have been used. The extension of point properties to curves appears in the form of principal patches [93], surface primal sketch [7, 19], planar curves and asymptotes [19]. Reconstructing surfaces from arbitrary surface curves have used the formulations of Gordon-Coons patches [33] or tensor-product surfaces [56], while some of the methods to reconstruct surfaces from intrinsic surface curves were studied in [19].

The interaction of scale and curve features has also been studied [7, 36]. In these representation methods, the surface is always described by uniform coverage with computation applied to every part of the surface. Hence, it is not possibly to represent the surface "incrementally."

Computation of Surface from Curvilinear Properties The desirable goals of representing 3D surfaces using intrinsic geometric properties that are also perceptual meaningful prompts us to look into computational procedures that link these two goals together. This investigation also provides a natural extension from 2D curve representations to 3D.



(a) Face with shading information.

(b) Face with apparent contour only (for a Lambertian surface).

Figure 1.4: The information conveyed by shading and apparent contour.

The way a 3D surface is projected onto the image plane is dependent on the light source, surface geometry and the projection process. Inversely, these factors can be effectively inferred if various information from visual modules can be integrated [4, 99]. From this point of view, we can talk about the "amount" of information available for the observer to compute surface geometry, if the light source and projection process are known or assumed. When only a single module is used to compute the surface, the available information is, in a sense, "minimal" and the problem is generally harder in this case. In this research, we specifically consider Lambertian surfaces, where the shape can be computed from extremal contours only, and textured surfaces, where optical flow is available for the surface (Figure 1.4). In the latter case, an active observer can isolate the foreground from the background using optical flow even when both the object and background are similarly textured.

One specific kind of extremal contour, the occluding contour, (or apparent contour) can be used to effectively constrain the surface [10]. Subsequently, the surface can be recovered from occluding contours by observing the contour "sliding" across the surface as the vantage point changes [19, 25, 39, 102]. On the other hand, stationary contours (curvilinear markings on a surface) and discontinuous contours only constrain the surface along a single dimension like a strip for a smooth surface [60]. Consequently, stationary contours have been studied mostly in the context of qualitative surface characterization [26, 59, 96, 117].

1.4.6 Active Vision, Optical Flow and Navigation

Given a task, the data necessary for performing the task depends on the task itself. So, for example, collision avoidance does not require surface shape recovery. On the other hand, many problems in vision are ill-posed or not well constrained because of the lack of sufficient data. These two cases indicate the close interplay between a problem and relevant data in computer vision. A balance can be struck if the observer is allowed to interact with the scene by continuous observation, i.e., the current observation is analyzed based on the nature of the task and data already acquired. These two aspects constitute the *active vision paradigm* [5].

Methods that seek additional data in the environment determined by the problem generally involves relative motion between the scene and the observer. This relative motion could come from either voluntary observer motion or the dynamical nature of the scene. In either case, the additional data made available by these processes are either the motion field computed from optical flow when the surface is textured, or the deformation of smooth surface features and features such as occluding contours.

Optical Flow Optical flow conveys information about a surface through depth cues. This information may be used to identify object boundaries (optical flow segmentation) [2], to determine the observer's egomotion (passive navigation) [51, 21], and to recover surface geometry. Biologically, optical flow can be modeled by filtering the visual signal at different time intervals and making appropriate comparisons [47, 101]. This approach is readily extended to filters of spatio-temporal receptive fields [1, 42, 43, 108].

The computational domain of optical flow is a spatio-temporal volume of the images. Since the motion field conveyed by the optical flow is relatively simple and involves only locally rigid motion such as rotation and translation, the result is easier to interpret than those computations in the spatial domain only (e.g., edge detection). This simplification comes from the *optical flow constraint* [51] which states that, for a locally rigid motion, the optical flow generated by a stationary point on an object will be smooth relative to its neighborhood if the neighborhood is entirely embedded in the same surface. This property has been applied globally to estimate optical flow. However, the condition where this equation holds breaks down frequently with an image of natural origin, and the problem of computing the motion field from optical flow is itself an ill-posed problem (e.g., due to the aperture problem [101]). On the other hand, when certain natural conditions of the surface are assumed (for example, finely textured), there is a well-defined correspondence between the motion field and optical flow, and the computation can be carried out effectively using spatio-temporal filters [45, 46].

Optical flow itself provides depth cues of the scene [74], and its segmentation generally corresponds to object boundaries (the apparent contour). However, there is a close interplay between the computation and the segmentation of optical flow, which may be separated to some degree by using iterative methods [51] or multiple scale filtering [110]. Optical flow also conveys useful information regarding surface geometry [62, 64]. When the spatial and temporal derivatives can be computed reliably, the surface geometry can be also recovered [109, 97, 100]. On the other hand, for navigation tasks that do not require full scene recovery, optical flow analysis is useful in exposing the relationship between the observer and the scene [17, 24].

Segmentation of an optical flow field is qualitatively similar to segmentation of image irradiance. The goal is to locate one-dimensional boundaries that correspond to actual object boundaries.

Deformation of Curvilinear Features For an active observer, when the surface is smooth and uniform in shading, the only reliable information about surface shape comes from the projection of occluding contours onto the image plane, which usually coincides with *silhouettes*. On the other hand, object surfaces often contain stationary curvilinear features (or surface markings). Both stationary and occluding contours are curvilinear features on the object surface and they constrain surface shape in a similar way, i.e., they tell us something about the tangent direction and degree the surface curves away from this direction. However, for a stationary observer, these two kinds of contours appear to be locally identical and, therefore, cannot be distinguished. On the other hand, the fact that an occluding contour slides across the object surface while a stationary contour is fixed on the surface present themselves quite differently to an active observer. This observation also makes the task of classifying contours an important problem for strategies that infer surface shape from contour [25, 69, 117].

For occluding contours, an active observer can recover surface shape from *known* occluding contours under orthographic projection [39] or arbitrary observer motion under perspective projection [25]. The same method also provides a procedure to identify the type of contour after the surface shape is recovered. However, this identification method requires accurate measurement as well as the recovery of the surface shape as a prerequisite. An alternative method is an affine-invariant based *re-projection* approach [69], though surface recovery using this method is limited to areas where the surface shape contains occluding contours. In summary, methods used to recover surface shape from occluding contour require accurate measurement of observer motion, camera calibration, and is, in general, very sensitive to noise. The advantages of these methods are twofold: contour features can be extracted more easily and reliably, and they can strongly constrain the surface shape and characterize the surface directly (e.g., how occluding contours relate to sign of Gaussian curvature on the surface) [19, 59]. Active occluding contours have also been successfully applied to dynamic tracking of objects [57, 25].

For stationary contours, most results obtained are qualitative in nature, including the conjecture that parts of the surface shape from stationary contour deformation might be recovered [96]. The deformation of image contours can be studied in general terms by relating observer motion parameters to the deformation of image contours [23]. It can be shown, for example, how the sign of normal curvature can be determined from properties of projected image contours (e.g., inflection points) [25], and at least three views of a contour are required to distinguish between a space curve and an occluding contour [117].

1.5 Thesis Outline

The central issue investigated in this thesis is the systematic usage of geometric models in computer vision. This involves the development of a geometric language for the modeling of both visual perception and high-level cognitive tasks (e.g., object recognition) in the human vision system as well as the computation of essential components in the geometric language.

The biological basis of the human visual system and the mathematical modeling of receptive fields are presented in Chapter 2 as a foundation for the study. This is followed by the specification of a 2D and 3D geometric language in Chapter 3 to prepare us for the development of 2D geometric computation from receptive fields, which is given in Chapter 4. The results of local computation for 2D curves are extended and used as a basis for global curve representation and 2D object recognition in Chapter 5. This result is extended to 3D in Chapter 6 when we consider stationary contours on object surfaces. The need for an active observer and their navigation ability also becomes apparent here. The active capability is further strengthened by optical flow computation and segmentation, which is developed in Chapter 7. Finally, the different characteristics of localized movement needed for optical flow computation and global navigation needed for full surface recovery and representation are combined and studied in Chapter 8.

Chapter 2

Biological Basis and its Mathematical Modeling

Computer vision is at once both formal and experimental. In spite of the immense computing power of current technology, computational mechanisms that are comparable to the visual abilities of natural systems still elude us. In light of this, the prominent and relevant aspects of natural vision systems are examined first for their possible insights to our problems. The emphasis here is the local computational properties at the front-end of the processing layers. It is particularly interesting to observe how the spatial variation embedded in the continuous influx of light could be filtered and organized even at this foremost part of biological systems. These principles and their mathematical descriptions are the major topics in this chapter.

A biological system determines how to conduct local computations at the system front-end as part of the data selection process. This selection process also defines the nature and capabilities of the system itself. Insights to both the data selection process and formal properties can be obtained by studying systems whose capability we desire to emulate on computers.

In addition to understanding what kind of information is relevant to a set of tasks, the study of mathematical models of biological systems will also help us understand the formal nature of the data as part of "information" (interpreted within a pre-defined context) as opposed to "signal" (interpretation-free). This is essential for vision since at certain points we need to cease considering the data as a neutral signal free from any interpretation and start bringing in additional assumptions that not only constrain the problem but also make the problem context clear. This goal can only be achieved when the mathematical models of the vision front-end are well established for the desired tasks.

In this chapter the study of the physiological and mathematical models of the system at the level of photoreceptors establishes that the essential elements of the computation and the data selection process at the front-end are:

- overlapping computational units that have local scope in both spatial and frequency domains
- these units are parameterized by a scale parameter that has a non-uniform distribution across space
- the results of the computation are complete in signal space

The mathematical model thus established is also verified by predicting hyper-channels in the human vision front-end.

The organization of this chapter is as follows. The essential facts about the physiology of the human visual system are reviewed first as a prelude to formulating mathematical models for operations specially designed to examine the information contents of visual signal. The formal models are shown to be complete in signal space and examples in image coding are given. Following this, the distinction between signal and information is explicitly spelled out and several models for processing the information content are presented. Examples are also given as applications of a new model that conforms to the psychophysical data. The resulting mathematical language in this chapter and the expansion of it in the form of geometric languages in the next chapter will become the theoretical framework of this thesis.

2.1 Fundamentals of the Human Visual System

Among the structures and functionality in the human visual system, the front-end (including the photoreceptors and their adjacent layers) is better understood than other parts higher in the visual pathway. This and related areas in the visual cortex will be our main focus in the development of formal models.

2.1.1 Retina

The front-end of the human visual system is the optic transducer, or the retina. A schematic digram of the stratification of cells in retina is shown in Figure 2.1. Incoming light is first transduced directly by the photoreceptors, which can be classified into cone and rod cells. In addition to cones and rods, the retina contains four classes of cells: horizontal, bipolar, amacrine and ganglion cells. The basic inter-cell connections can be classified into an outer plexiform layer, where photoreceptors synapse with both horizontal and bipolar cells, and an inner plexiform layer, where bipolar cells synapse with both amacrine and ganglion cells. Finally, the axons of ganglion cells are bundled into the optical nerve and run through the blind spot to the lateral geniculate nucleus (LGN) and the visual cortex. If this signal transforming process is treated as an input-output system, then each output, i.e., the signal carried along the optic nerve of a single ganglion cell, can be defined by a *receptive field* (rf), which is the region of the retina, usually roughly circular, where afferent stimulation affects the overall firing rate of the output neurons. There is a large collection of literature treating the classification of receptive fields. Roughly, there are three types of rf: X, Y and W types. Each rf type is characterized by the effective area of the response and the way it responds to either a stationary or a transient input, as well as the responding speed.

It is important to observe that tradeoff of design is ubiquitous in various biological vision systems. The first example is the mechanism of color detection. It serves an extremely impor-

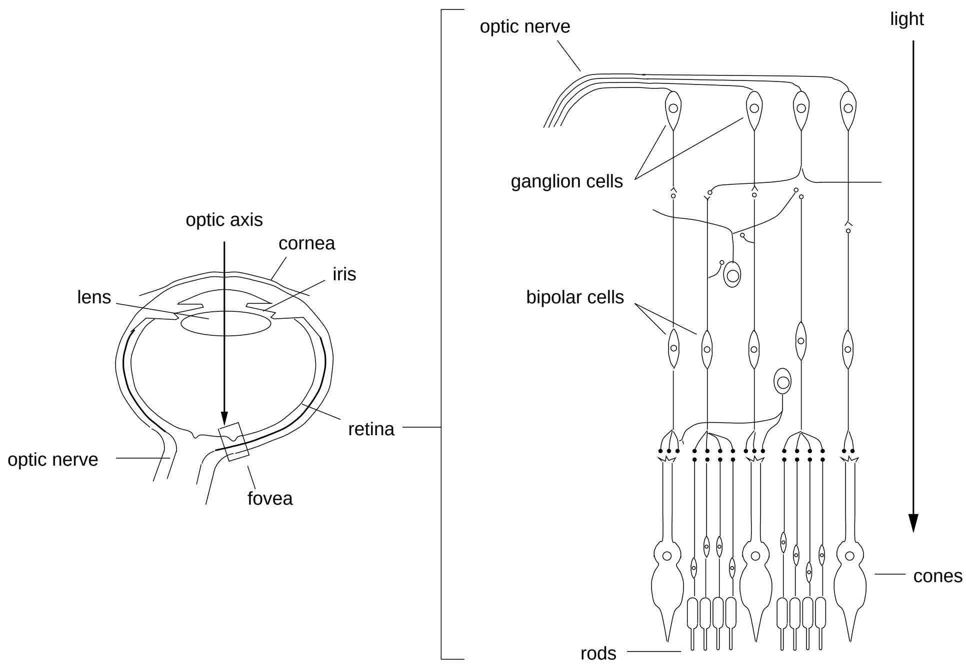


Figure 2.1: Structure of the eye.

tant role in the human visual system. However, a retinal cell that is sensitive to a particular color implies the loss of light energy associated with other parts of the visible spectrum. This in turn implies the failure of detection of moving objects in dim light. Consequently, for species whose optic sensors are used primarily for detection rather than recognition, it may not be worthwhile to have this capability. Besides humans, other color sensitive animals include the diurnal birds, reptiles, and octopus, while dogs, cats and frogs are color blind.

This compromise in sensitivity also occurs within the eyes of individual organisms. In the human retina, the cone cells are responsible for detailed vision and are color sensitive. The distribution of the cones is most condensed and uniform in the small area around the fovea. On the other hand, the rod cells are almost two orders of magnitude more sensitive to light than the cones and are incapable of detecting color.

It is worth noting that there are about 126 million photoreceptors in each eye, of which 120 millions are rod cells and 6 million are cone cells, but there are only 1 million nerve fibers exiting each eye. Hence, on average, the information compression rate is about 126:1. However, neither the distribution of photoreceptors along the surface of the retina nor the distribution of ganglion cells rf with respect to the photoreceptors is uniform. In terms of the theory of signal processing, we have two sampling schemes, one is the sampling of incoming light by the mosaic of photoreceptors and the other is the equivalent sampling of incoming light by the receptive fields, which is achieved through the inner and outer plexiform between ganglion cells and photoreceptors. Schematic diagrams for the sampling mosaic of photoreceptors (cones) and the equivalent rf are shown in Figure 2.2 .

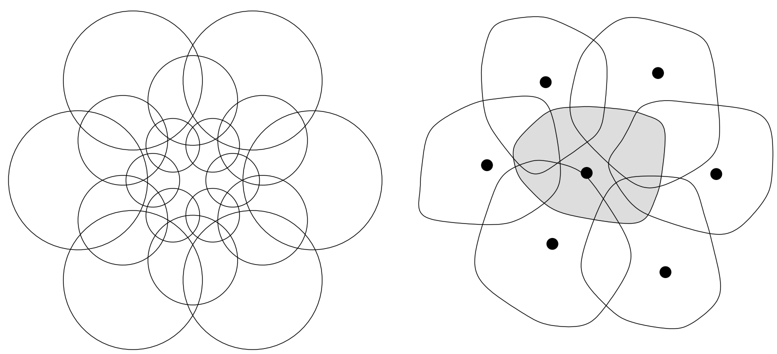


Figure 2.2: Sampling topology of the retina.

The layout of optic receptors has maximum uniformity and density at the center of the optic axis of the eye, and decreasing density and regularity with increasing eccentricity. In the human fovea, cones are spherically packed in a hexagonal array and there are almost a corresponding number of ganglion cells connected to these cones. However, there is an intense overlapping of the receptive fields [71]. It was argued by Levick [71] that the notion of different types of receptive fields overlapping a common area leads naturally to a parallel pathway design of the visual system. The property that there are almost equal numbers of rf and corresponding cones implies, at least theoretically, the lossless information processing from the fovea to the optic nerves. Overlapping is depicted in Figure 2.2 as part of the sampling mosaic.

One of the basic requirements for a system ensuring high spatial resolution is regular spacing of the sample points. Irregularly spaced sample points introduce positional noise and reduce visual acuity [104]. The mosaic of ganglion X cells shows neither square nor hexagonal symmetry, but the distances between cells seem to be regular [104].

2.1.2 Visual Pathway

The major stations visited by the optic nerve exiting from the retina are the lateral geniculate nucleus (LGN) and the visual cortex. The LGN is largely responsible for the relay of visual information to the visual cortex, which is roughly subdivided into three areas: areas 17, 18, and 19. The results of Hubel and Wiesel [52] showed that the so-called simple cells in area 17 are orientation selective according to the physical structure of "hyper columns." Each column is selective to only one direction. Later other results were established concerning the frequency selectivity of the simple cells (DeValois *et al.* [29]). Today, it is well acknowledged that simple cells are characterized mainly by these two properties — orientation and frequency selection. It is also established that the response of simple cells is roughly linear [27, 79]. This property of linearity is the basis for a whole range of theories. There are, of course, highly nonlinear cells in the visual cortex. These cells are termed "complex" and "hyper complex" cells and are the main occupants of areas 18 and 19. There is no coherent theory about these areas except that they respond to more complex combinations of image structures.

As Daugman [28] explained, vision research has debated the basic functional organization of early visual representation since the 1960's. Both local feature detection in the spatial domain and frequency domain decompositions based on linear transformations such as Fourier analysis have been proposed. Marčelja [79] argued that if linearity holds for the cells being modeled, an adequate theory based on spectral transformation can be established and both domains are complementary to each other.

Some of mathematical models formulated along this line of thought will be reviewed in the next section, especially Gabor filtering [27, 79], and how these models can be used to distinguish the representation of visual "signal" and "information." A theory of image coding is also reviewed, which is motivated by this knowledge about the receptive field of the retina and the selectivity properties in frequency and orientation of simple cells in the visual cortex.

2.2 Retinal and Cortex Modeling

Our knowledge of the front-end mechanisms in the visual pathway, as reviewed in the previous section, will be formalized in this section and serves as a foundation for the subsequent chapters. The emphasis is to show how the narrow exit of the visual front-end and local computations as performed by the physiological mechanism of the exiting information path can be described elegantly by mathematical formulations.

2.2.1 Modeling Using Gabor Filters

Gabor [38] proved that for a given temporal signal, the degree of simultaneous localization in both time and frequency domains, which is measured by the multiplication of the standard deviations of the signal in both time and frequency domains, is lower-bounded by $1/4\pi$. He also derived the general form of the signal that actually achieves this lower bound. This signal form is known as the Gabor function :

$$\psi(t;\mu,\sigma,\lambda) \triangleq \exp\left[-\frac{(t-\mu)^2}{2\sigma^2} + i\lambda t\right] \tag{2.1}$$

and has the Fourier transform:

$$\Psi(\omega;\lambda,\sigma,\mu) = (2\pi)^{1/2}\sigma\exp\left[-\frac{\sigma^2(\omega-\lambda)^2}{2} - i(\omega-\lambda)\mu\right]. \tag{2.2}$$

If the effective spread in the temporal and frequency domains is defined as $\triangle t = \overline{(t-\bar{t})}^{1/2}$, and $\triangle\omega = \overline{(\omega - \overline{\omega})}^{1/2}$, then the Gabor signal satisfies the condition:

$$\triangle t \triangle\omega = \frac{1}{4\pi}.$$

It can also be shown that ψ is orthogonal in the sense

$$\iint_{-\infty}^{\infty} \psi(t;\mu,\lambda)\psi^*(s;\mu,\lambda)\, d\mu\, d\lambda = 2\pi^{3/2}\sigma \exp\left[-\left(\frac{t-s}{2\sigma}\right)^2\right]\delta(t-s). \tag{2.3}$$

Hence,

$$\iiint_{-\infty}^{\infty} \psi(t;\mu,\lambda)\psi^*(s;\mu,\lambda) f(s)\, d\mu\, d\lambda\, ds = 2\pi f(t). \tag{2.4}$$

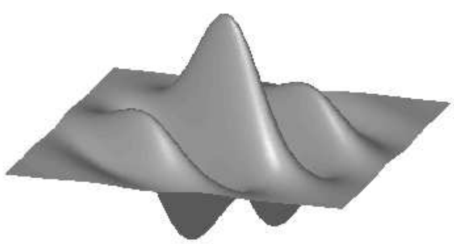


Figure 2.3: Two-dimensional Gabor signal in the spatial domain.

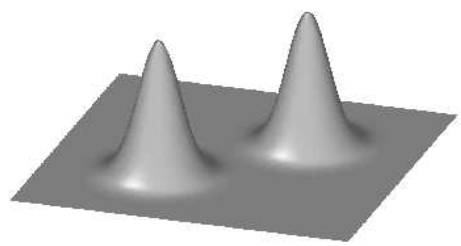


Figure 2.4: Two-dimensional Gabor signal in the frequency domain.

The traditional concept of the rf profile of a cell is a representation of the cell's response in terms of a bivariate function. It is defined as a weighting function, which describes the weighted contribution of light at each point in the receptive field to the response of the cell. It has been shown by Daugman [28] that the receptive field profile of the visual cortex simple cells can be adequately described by a two-dimensional Gabor filter (Figure 2.3 and 2.4):

$$\begin{aligned}\psi(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\lambda}) &= \psi(x_1, x_2; \mu_1, \mu_2, \sigma_1, \sigma_2, \lambda_1, \lambda_2) \\ &\triangleq \exp\left[-\frac{(x_1-\mu_1)^2}{2\sigma_1^2} - \frac{(x_2-\mu_2)^2}{2\sigma_2^2}\right]\exp[i\boldsymbol{\lambda}\cdot\mathbf{x}]\end{aligned} \tag{2.5}$$

or, equivalently, in Fourier form:

$$\Psi(\boldsymbol{\omega}; \boldsymbol{\lambda}, \boldsymbol{\sigma}, \boldsymbol{\mu}) = 2\pi\sigma_1\sigma_2 \exp\left(-\frac{1}{2}|\boldsymbol{\sigma}\cdot(\boldsymbol{\omega}-\boldsymbol{\lambda})|^2\right)\exp[-i\boldsymbol{\mu}\cdot(\boldsymbol{\omega}-\boldsymbol{\lambda})].$$

It can be observed that, geometrically, ψ is a Gaussian envelope centered at (μ_1, μ_2) and superimposed by a planar sinusoidal grating with spatial frequency $(\lambda_1^2+\lambda_2^2)^{1/2}$ and orientation $\theta = \tan^{-1}\lambda_2/\lambda_1$. The Fourier transform of ψ is a Gaussian envelope centered at (λ_1, λ_2) and superimposed by a sinusoidal grating with spatial frequency $(\mu_1^2+\mu_2^2)^{1/2}$. Also, the standard deviation along axis ω_1 (ω_2) is the inverse of the standard deviation along axis x_1 (x_2) (see Figure 2.5).

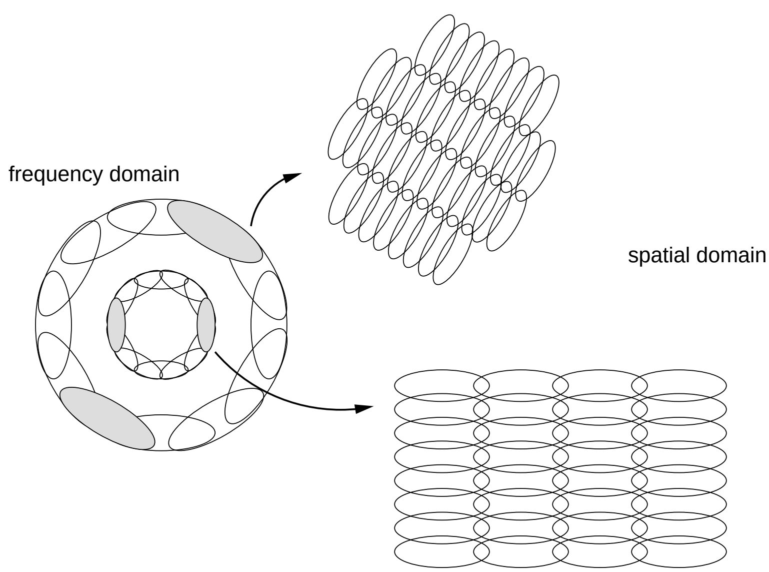


Figure 2.5: Relationships of Gabor filters between spatial and frequency domain.

If a receptive field is characterized by the profile $p(x, y)$, then its response to an input image

$f(x, y)$ is

$$r = \iint_{-\infty}^{\infty} p(x, y) f(x, y)\, dx\, dy. \tag{2.6}$$

If $p(x, y)$ is a prototypical profile at (0, 0) of the image plane, then, for a set of receptive fields at (x_i, y_i), we can acquire the following set of responses:

$$\begin{aligned} r(x_i, y_i) &= p(-x, -y) * f(x, y)|_{x=x_i, y=y_i} \\ &= \mathcal{F}^{-1}\{P(-u, -v)F(u, v)\}|_{x=x_i, y=y_i}, \end{aligned}$$

where P and F are the Fourier transforms of p and f, respectively. Hence, the response of the rf at (x_i, y_i) would be the sample value of $\mathcal{F}^{-1}\{P^* F\}$ at (x_i, y_i) if p is real.

An actual set of Gabor filters designed by Watson [105] includes eight filters with spatial frequencies spanning from 0.25 to 32 cycles/degree; each has a bandwidth of one octave. The orientation of the filters at a specific frequency includes ten filters with orientation spanning from 0 to 360 degrees and each has a bandwidth of 36 degrees.

For a specified Gabor profile, the most useful property is apparently that the center of the profile in the frequency domain characterizes the properties of selectivity of both spatial frequency and orientation of the profile. Furthermore, both the orientation and the spatial frequency bandwidth of the profile are highly dependent on the shape of the profile as well as the center frequency. However, from data supplied by psychophysics [29, 30] and psychology [28], the visual system of the macaque monkey has the following characteristics: $\frac{1}{4} < \sigma_1/\sigma_2 < 1$, the median frequency bandwidth is about 1.4 octaves, and the median for orientation full bandwidth is about 40° (the frequency range of human perception is about 0 to 60 cycles/degree). Based on these data, the stability of the orientation half-bandwidth can be guaranteed by the approximate relation $(\lambda_1^2 + \lambda_2^2)^{1/2} \gg \sigma_1, \sigma_2$.

2.2.2 Signal and Information Representation

2.2.2.1 Signal and its Representation

A signal is defined with respect to its source. Hence it is desirable to preserve as much as possible the original signal when devising a representation for it. As a consequence, the ideal signal representation process is a bijective mapping between two Hilbert spaces, in which the destination of the mapping from the source signal is called the representation of the signal, and vice versa. The mathematical properties of the representation depend to a significant degree on the choice of the bijective mapping. In this section, three kinds of mappings on a well-behaved (commonly equivalent to the concept of smoothness) two-dimensional signal will be compared. Among them, two are uniform Fourier transforms on the whole Euclidean plane coordinated by Cartesian and polar coordinates, respectively, and one is a localized Gabor transform. It is shown that, on the L^2 Hilbert space, there can be defined a corresponding *complete* set of basis signals for each mapping.

Fourier Transform in Cartesian Coordinates Given a signal $f(x, y)$ in the 2D Euclidean space with Cartesian coordinates x and y, the Fourier transform pair is defined as

$$F(u, v) \triangleq \mathcal{F}\{f(x, y)\} = \int_{-\infty}^{\infty} f(x, y)e^{-i(ux+vy)}\, dx\, dy$$

and

$$f(x, y) \triangleq \mathcal{F}^{-1}\{F(u, v)\} = \frac{1}{4\pi} \int_{-\infty}^{\infty} F(u, v)e^{i(ux+vy)}\, du\, dv.$$

Fourier Transform in Polar Coordinates If the polar coordinates in $x - y$ and $u - v$ planes are defined as

$$\begin{cases} x = r\cos\theta \\ y = r\sin\theta \end{cases} \qquad \begin{cases} u = \rho\cos\phi \\ v = \rho\sin\phi \end{cases}$$

then $f(x, y)$ can be transformed into $f(r, \theta)$ and expressed as the expansion of a Fourier series:

$$f(r, \theta) = \sum_{n=-\infty}^{\infty} C_n(r)e^{in\theta}.$$

The Fourier transform of $f(r, \theta)$ is

$$F(\rho, \phi) \stackrel{\triangle}{=} \mathcal{F}\{f(r, \theta)\} = \sum_{n=-\infty}^{\infty} 2\pi\overline{c}_{nn}(\rho)e^{-in\phi}$$

where

$$\overline{c}_{nn}(\rho) = \int_0^{\infty} rC_n(r)J_n(r\rho)\,dr$$

is the n-th order Hankel transform of $C_n(r)$.

Gabor Transform For the Gabor transform $\psi(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\lambda})$ with $\sigma_1 = \sigma_2$ being constant over the space $(\boldsymbol{\mu}, \boldsymbol{\lambda})$, the transform of $f(\mathbf{x})$ is defined by

$$F(\boldsymbol{\mu}, \boldsymbol{\lambda}) \stackrel{\triangle}{=} \iint_{-\infty}^{\infty} \psi(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\lambda})f(\mathbf{x})\,d\mathbf{x} \tag{2.7}$$

and it can be shown that the inverse transform is

$$f(\mathbf{x}) = \frac{1}{4\pi^2}\iiiint_{-\infty}^{\infty} F(\boldsymbol{\mu}, \boldsymbol{\lambda})\psi(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\lambda})\,d\boldsymbol{\mu}\,d\boldsymbol{\lambda}. \tag{2.8}$$

It is well known that the discrete version of the Fourier transform in either Cartesian or polar coordinates is complete and the coefficient corresponding to each basis function can be obtained in an elegant manner [94]. On the other hand, the coefficients of the discrete version of the Gabor transform, while complete, requires solving a set of linear equations [79]. This is due to the pseudo orthogonal property of the discrete Gabor kernels.

2.2.2.2 Information and Second-Generation Image Coding

Information is meaningful only when the purpose of using the information and, hence, the information itself is clearly defined. When images are encoded for the purpose of visual perception, the criterion for a successful representation of the information can only be judged under the scrutiny of human eyes. In this section, theories and results from Kunt *et al.* [68] and Watson [106] are presented in the context of information representation. Following this, an operational model for human visual information encoding is considered, which, instead of using uniform filtering over the entire image plane like those of Kunt *et al.* and Watson, makes use of a sparse coding. The concept of sparse coding has a natural correspondence in the physiology of the human visual system [20], as will be described later.

It is known that under some specific conditions such as band limitation, the discrete representation of a continuous signal is perfect. Furthermore, if there is no a priori knowledge about an image, the canonical signal representation of the image would be the sampled locations in the two-dimensional space along with the quantized amplitude at each location. The purpose of image coding is to reduce the amount of storage needed to reconstruct the original image. This is possible only when redundancy exists in the picture. Traditionally this is achieved through information theory, i.e., using a signal processing approach to select a string of messages that characterize the original image and then applying source coding theory to code the messages in either a lossless or lossy manner. The techniques of processing may either be spatial in nature (such as PCM, predictive coding), or characterized by transformational methods (such as Karhunen-Loève transform or Fourier transform). In the review paper by Kunt *et al.*, it is observed that the compression ratio of these so-called “first-generation coding techniques” has reached a saturated value of 10:1 in recent years. In the same paper the “second-generation coding techniques” were described, which originate from studies of brain mechanisms of vision. What differentiates these two approaches to image coding is attributed basically to the different points of view concerning “information” and “information loss.”

It is known from both physiology and psychophysics that in general the response of various biological cells to a sustained input rapidly fades away. On the other hand, within cells there are two major categories. One is characterized by a longer persistence of response and the other is more transient in nature. The existence of mutual inhibition between these two categories [20] implies that the parts of an image that are primarily textural and the parts that are primarily consists of contours may have different information content. This leads to the so-called “contour-texture” techniques. It can be argued that the transient parts of the afferent signal arouse attention and demand higher visual resolving capability. This is where the contours become essential. On the other hand, the significant difference between a line drawing picture and its fully textured version hints at the role played by the texture. This “contour-texture” or “high-frequency, low frequency” technique was implemented by Hunt *et al.* as two frequency bands along with a bank of directional filters.

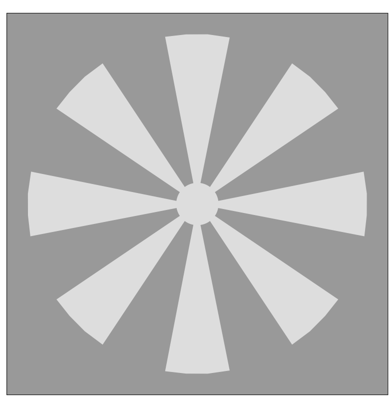


Figure 2.6: A set of directional filters decomposed into high and low pass components.

Let a set of high pass filters be defined (Figure 2.6) with the response

$$H_i(u, v) = \begin{cases} 1 & \text{if } \phi_i < \tan^{-1}(v/u) < \phi_{i+1} \text{ and } (u^2 + v^2)^{1/2} > \rho_c, \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

with

$$\phi_i = (i - 1)\pi/2n$$
$$\phi_{i+1} = (i + 1)\pi/2n,$$

where ρ_c is the cutoff frequency of the low pass filter (hence, the high pass filters). If an image is filtered by H_i and the corresponding low pass filter and the high band zero-crossing is detected and encoded for both its magnitude and location, then the original image can be reconstructed by combining the low pass result and the interpolated high pass result.

Watson [106] used a set of "cortex transforms" to explore this scheme of coding in the context of human visual perception. The cortex transform is designed to cover the whole frequency plane. Special techniques such as quantization using contrast masking and sub-sampling of image layers by the cortex transform (most of the layers have a bandwidth smaller than the original image) were also used.

Watson argued that a biologically feasible coding is achieved by the fixation of the eye, that is, stabilizing the image on the retina and fixing the fovea at a particular point. He also maintained that to extend the resolving capability of the fovea to the entire image plane, which is his case, has practical usage in image compression. It may not be evident what the difference is between the encoded information for recognition purposes and for visual communication purposes. However, due to the diversity exhibited in individual perception across all human beings, the information for communication might be more than what is necessary for individual use.

On the other hand, a 2D Gabor filter (Eq. (2.5)) is indexed by three set of parameters, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\lambda}$. If this filter is expanded to cover the full image plane for fixed $\boldsymbol{\sigma}$, then this can be termed a cortex filter as in Watson's sense. If we design the filter so that the parameters $\boldsymbol{\sigma}$ are indexed by eccentricity, the filter can then be adequately called a *retina filter*. Watson argued in [107] that this kind of coding can be useful only for one fixation. Nevertheless, single-point fixation coding with adequate selection of $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, $\boldsymbol{\lambda}$ might prove to be more economical for image coding in the context of human vision. This kind of coding can be named "sparse coding." However, three problems must be resolved before this coding can be really beneficial. The first difficulty is to be able to delineate the object boundary by its contour. This requires at least the capability of being able to analyze texture and color. Stereopsis is helpful but not necessary. The second problem is a pre-analysis of the image is required, which serves the purpose of choosing the fixation point and then the retina filtering can be used to encode the image. The last problem is, in the case of complex objects which cannot be decomposed, to establish a temporal association mechanism for the coding of multiple fixations.

2.2.3 Channel Models of Receptive Fields

The incoming visual signal for a biological observer is necessarily distributed across both physiological structure and information space (e.g., the space of retinal filters). Consequently, it will be beneficial to explore this distributed structure in information extraction and filtering as well as the uniformity of the computational process..

For biological systems, the classic work of Hubel and Wiesel provides evidence of orientation distribution in visual signal processing. Combined with other work on spatial frequency selectivity, the distribution of visual information in 2D orientation-frequency space has become a research topic. It has been proposed that along with other quasi-independent major modules in early vision systems, the orientation-frequency selectivity module serves important roles in edge detection, attention control, and visual memory coding.

For a Gabor filter, the sinusoidal grating serves as a shifter in the frequency domain. There are other filter forms similar to sinusoidal grating such as the Marr's Laplacian of Gaussian, Wilson and Bergen's DOG channel model, and Canny's optimized edge operator [22]. They all have the similar form of interleaving excitatory and inhibitory regions (Figure 2.7). The difference is the number of interleaving regions. Hence, a more general form of orientation-frequency selectivity is

$$\psi(x_1, x_2; \lambda, \theta) = G_\sigma(x_1, x_2) S(\lambda, \theta)$$

where G_σ is the envelope with finite support controlled by σ and $S(\lambda, \theta)$ is an orientation selection function of the spatial frequency λ and the orientation θ.

The specific form of $S(\lambda, \theta)$ depends on other constraints on the desired visual information. A particularly useful one can be derived by observing that in the psychophysics-based data of Wilson and Bergen's channel model, the ratio of excitatory area to the difference of excitatory and inhibitory area is approximately zero (9.4×10^{-3}) for the two highest channels, N and S, and hundreds of times higher for channels T and U. In reality, the total area of $S(\lambda, \theta)$ controls the contrast sensitivity of adjacent areas. Hence it is more appropriate using Wilson and Bergen's data in higher channels.

Wilson and Bergen's channel model [113] has the explicit form:

$$S = C_i(x) \triangleq \exp\left(-\frac{x^2}{\sigma_i^2}\right) - c_i \exp\left(-\frac{x^2}{(b_i \sigma_i)^2}\right) \tag{2.10}$$

where i is indexed by the channel N, S, T and U. Hence the orientation-frequency selection function $G_\sigma S$ is

$$\exp\left[-\left(\frac{-x_1 \sin\theta + x_2 \cos\theta}{\sigma}\right)^2\right] C_i(x_1 \cos\theta + x_2 \sin\theta)$$

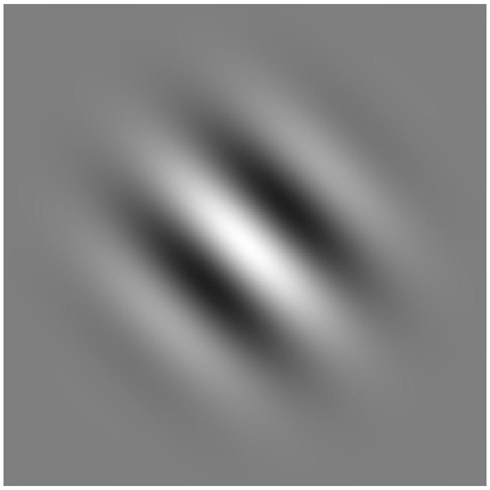


Figure 2.7: Prototypical visual sampling cell in both spatial and frequency domain.

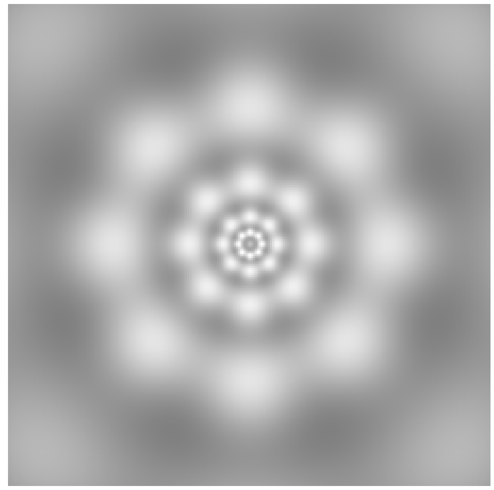


Figure 2.8: Visual sampling by five bands Gabor cells.

or in its frequency domain form:

$$\exp\left[-\pi^2\sigma^2(-u\sin\theta + v\cos\theta)^2\right] \cdot \left[\exp\left(-\pi^2\sigma_i^2(u\cos\theta + v\sin\theta)^2\right) - b_i c_i \exp\left(-\pi^2 b_i^2\sigma_i^2(u\cos\theta + v\sin\theta)^2\right)\right].$$

The constraint mentioned above is equivalent to $b_i c_i = 1$.

In order to actually maintain a quasi-circular receptive field, the value of σ should be determined from $C_i(x)$ by fitting a Gabor filter to $C_i(x)$. This process also determines the approximate spatial frequency of $C_i(x)$, which is 9.8, 5.0, 2.6 and 1.5 for N, S, T and U, respectively. However, as Marr *et al.* [78] pointed out, a channel with higher spatial frequency should exist and the diameter of the central excitatory region is predicted to be between 1' and 2'. Using the aforementioned constraint for the high frequency channel and extrapolating the spatial frequency to 20 (as predicted by the arithmetic sequence of 2.6, 5.0 and 9.8), the "smallest channel" can be derived with the parameters $b_i = 1.604$, $c_i = 0.623$, and $\sigma_i = 0.014$. This channel has a central excitatory region with diameter around 1.5 minute, in conformity with Marr *et al.*'s prediction.

Though the temporal response is essential for attention control, it is reasonable to conceive that the degradation of resolution with increasing eccentricity serves the purpose of establishing a visual information hierarchy in a static context, as Burt's Laplacian pyramid does. Bearing this in mind, a further step taken is to assess the effectiveness of the approach in the extraction of essential visual information, such as edges.

Figure 2.9: Original gray level image of a house.

Figure 2.10: ***H*** **channel information of the house.**

As is explicated by Kunt *et al.* [68], these biologically motivated coding techniques can achieve an image compression ratio of more than 50 or even 90. However the issue which addressed in their paper is the reconstruction problem, and as far as cognition is concerned, the focus of computation is inevitably shifted to somewhat different properties such as edge information extraction and fixation control. To illustrate this, Figure 2.8 is used as a model for the layout of visual information sampling in the frequency domain. Due to the duality between the spatial domain and the frequency domain, the layout also serves as a model for the spatial distribution of sampling cells within the visual field. The modeling is based essentially on the NSTU model of Wilson and Bergen plus the hyper-channel derived above, and will be called the HNSTU model. However, the computation is partly implemented by a general Gabor scheme; that is, the other dimension of the HNSTU model is fit into two-dimensional Gabor form.

Figure 2.11: N channel information of the house.

Figure 2.12: S channel information of the house.

The flatness of the response of the layout can be observed by actually operating each channel independently on a test image and recovering the image by arithmetical summation. In the experiment a uniform sampling aperture across the visual field is assumed; hence no actual degradation is observed with respect to the visual fixation point. By optimizing the flatness of the Gabor covering, the standard deviation of the profile is around 0.28 of the central frequency, which implies that the spatial overlap between adjacent cells is controlled to the extent that it conveys the same amount of information as the center of each cell does. This is somewhat smaller than the result obtained by fitting the HNSTU model to the Gabor profile, which is around 0.42 of the central frequency. The additional overlap is related to the compensation of computation across the boundary of cells during the optimization, i.e., the diminishing response toward the cell perimeter is compensated by adjacent cells.

On the other hand, the crucial aspect of the visual information hierarchy is to explore the information contents at various resolutions. The result obtained by the additional consideration of degradation of resolution with increasing eccentricity is shown in Figure 2.13 and 2.14. One of the fixation points is chosen to be the position with maximum response to the computation of edge information, and the other fixation point is chosen at a corner point. In both cases, the physiological data of approximate double the size of visual sampling cells for every 4° of eccentricity is assumed. Also, the extent of the visual field is taken as 20.5°, which is based on the results of psychophysics.

Figure 2.13: Girl image with fixation center around eyes.

Figure 2.14: Girl image with fixation center at lower-left corner.

2.3 Summary

A data-processing system involves both the data model and the processing model. In this chapter we looked into some of the processing models built into biological systems. The insights provided by the biological construction hint at how certain operations are beneficial in analyzing the information carried by the incoming light. The formalization of early stages in this analysis is also done in this chapter following the presentation of some of the unifying principles in biological systems. The mathematical formulations thus acquired is not only a faithful model for biological systems but also a foundation upon which crucial geometric information can be extracted and manipulated.

Data models in a visual processing system will be covered in the next chapter. These include models for images, one-dimensional curvilinear features in an image and on the surface of an object, and three-dimensional surfaces.

Chapter 3

Image, Contour, and Surface Modeling

The most fundamental data in a vision system are images. Further up in the data abstraction, there are curves and surfaces that provide geometric descriptions of the images. In this chapter multiple models of these fundamental geometric concepts are presented according to different contexts, since the relevancy of representation and models is tied closely to the tasks to be handled.

The essential characteristic of the processing models in the previous chapter is the biological functionality of natural systems. In this chapter, however, the emphasis is on the formal properties of the data models of images, curves and surfaces as well as their perceptual foundation.

The task that most interests us is the manipulation of objects. This requires us to define a shape description language that is both geometric in nature (so that spatial relationships between objects can be measured) and invertible by computational processes, i.e., the specification of the language needs to be operational in the sense that a natural computational procedure is associated with each element in the language. This not only enforces the computational nature of the problem but also dictates a structural hierarchy starting from an image all the way up to 3D shapes that are to be manipulated.

The sampling mosaic of the photoreceptors provides us with a way to restore the continuous form of the image so that a differential geometry language can be used to describe the required geometric structure for a given task. Several formal models of curves and surfaces are presented in their analytical and computational form so that they form a computational substrate for the following chapters. It is also shown how perceptual features can be mapped to geometric features.

The chapter begins by applying the theory of sampling to images so that its continuous form can be restored and differential operations can be applied with known effect due to the sampling noise. Two-dimensional curves are treated next. Several formulations for modeling curves are presented in a logical way which leads finally to a representation in geometric feature space using Hermite functions. Finally, surfaces are considered from the point of view of triangulated patches and principal curvatures. The former is a natural formulation when an observer tracks points on a textured surface, while the latter is derived from recovering curves on the surface.

3.1 Image Models

A digital image is usually modeled in a discrete 2D domain by $I(x, y)$ with x and y taking only integer values within the range $(0, N-1)$. This discrete model will have to be somehow rectified if we need to establish a mathematical foundation upon which both geometrically meaningful and differential structures are to be organized.

A natural way of representing images for the purpose of establishing a continuous structure is to fit the discrete grid to a continuous model so that gaps between points can be "filled" smoothly. However, the continuous data model has to satisfy some "well-defined" conditions and support operations that are essential for manipulating the geometric structures. Historically, this is achieved by representing the image redundantly as a parameterized family, in which each member has a definite relationship with other members of the family and can be derived from a common source—the *image generator*. Pyramids, multi-grid and scale-space are examples upon which this family can be built.

3.1.1 Image Generator

Theoretically, an image is a continuous signal in a two-dimensional space. The physical images available for processing in both biological and artificial systems are derived from this continuous source by a sampling process. The well-known sampling theorem informs us that the representation of an analog signal by a digital version is reversible if the sampling rate is higher than the *Nyquist rate*. That is, full content of the analog signal can be recovered from the sampled version providing the sampling rate is higher than the Nyquist rate.

By following the rule of *circular convolution*, all computations in the analog domain can be executed equivalently in the digital domain up to the precision of the computational process. Hence the size of a digital image is constrained by the Nyquist rate and all the information is contained within a bandwidth of $(-B, B)$, where $B = \pi$. The bandwidth is expressed in angular frequency with sample interval $T = 1$ [83]. However, this equivalence is only valid if the continuous signal is defined completely, a situation that is never true in the real signal. Consequently, we will have to look into other kinds of equivalence for our computational purpose.

Let's consider an imaging process that is directly responsible for converting the continuous irradiance into a sampled version and all subsequent processing is done on this sampled signal. We will denote this initial sampled signal $I(\mathbf{x}) = I_0(\mathbf{x})$ and name it the *image generator* or just *generator* (called *inner scale* in [37]). $I(x, y)$ can be viewed as corresponding to the finest resolution available from the optical sensor (e.g., retina) [58]. As discussed above, if $I(\mathbf{x})$ is transformed into the frequency domain, each of its samples cannot be more than π/N apart, where N is the number of samples in one dimension.

Since properties important to visual perception are intrinsically geometric, these properties are necessarily invariant to rotational and translational coordinate transforms. The local geometric properties can be naturally acquired through differential geometry. However, since differentiations of various order are involved, conventional techniques for differentiating images are not very useful because of their numerical instability. This instability can be eliminated if we never compute differentials on the generator $I(\mathbf{x})$ (see the convolution property of the Gaussian kernel in the next section). This assumption can be readily justified if we adopt the convention that we always make use of an imaging device with a resolution higher than the highest resolution that we will ever need for the computation.

3.1.2 Image Representation Using Gaussian Kernels

Gaussian filters are ubiquitous in natural visual systems and are used traditionally for modeling image blurring. At the pixel level, any filtering operation using a Gaussian kernel will result in information loss and a lossless representation of an image will have to include all sizes, starting with a physically determined lower bound $\sigma_{\min}$ (see Section 4.3.1). This is because the information in an image that is derived by convolving the generator with a Gaussian kernel of a particular size always contains the intensity information of images that is derived by convolution with Gaussian kernels of larger sizes. Henceforth, images computed by convolving with Gaussian kernels will all be redundant except the generator, since all the information can be generated from the generator alone. However, as we will see shortly, there are important computational reasons for this redundancy.

A one-dimensional Gaussian kernel is defined as

$$\psi_0(x;\sigma) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp(-\frac{x^2}{2\sigma^2}) \tag{3.1}$$

and a two-dimensional kernel is defined as

$$\psi_{00}(\mathbf{x};\sigma) = \psi_0(x;\sigma)\psi_0(y;\sigma). \tag{3.2}$$

Let the i-th order differential of $\psi_0(x;\sigma)$ be denoted by $\psi_i(x;\sigma)$. Since the two-dimensional Gaussian kernel is separable, we have

$$\psi_{ij}(x, y; \sigma) = \psi_i(x; \sigma)\psi_j(y; \sigma). \tag{3.3}$$

These filters will be referred as *receptive fields* henceforth (its biological counterpart is described in Section 2.1.1). The similarity of their forms and response profiles of receptive fields in biological systems have been observed (see e.g., [66]). The first-, second- and third-order differentiation of ψ_0 are given by

$$\begin{aligned}
\psi_1(x;\sigma) &= -x\psi_0(x;\sigma)/\sigma^2 \\
\psi_2(x;\sigma) &= -(1 - x^2/\sigma^2)\psi_0(x;\sigma)/\sigma^2 \\
\psi_3(x;\sigma) &= x(3 - x^2/\sigma^2)\psi_0(x;\sigma)/\sigma^4
\end{aligned} \tag{3.4}$$

These basis functions are depicted in Figure 3.1.

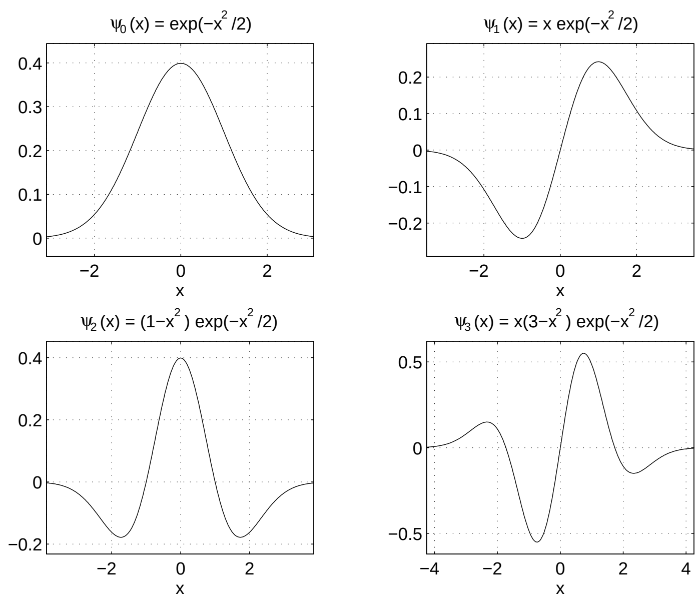


Figure 3.1: ψ kernel and its 1st, 2nd, 3rd-order differentiation.

These filters have some important properties that are crucial for our computation (see [66]), especially the *convolution property*:

$$\frac{\partial^{i+j}}{\partial x^i \partial y^j}[\psi_{00}(x, y; \sigma) * I(x, y)] = \psi_{ij}(x, y; \sigma) * I(x, y) = \psi_{00}(x, y; \sigma) * I_{ij}(x, y). \quad (3.5)$$

It is this convolution property that allows us to eventually replace differential operations on the image by integral operations when the image is parameterized in a scale-space and avoid the problem of noise amplification by differential operations. However this advantage comes with the expense of expanding the spatial range of computation as the order of differentiation increases.

An image sequence, called the *scale-space* of the image generator, can be generated from the generator and the Gaussian kernels, with σ being the *scale*, as follows:

$$I_\sigma(\mathbf{x}) = \psi_{00}(\mathbf{x}; \sigma) * I(\mathbf{x}) \quad (3.6)$$

The redundancy of $I_\sigma(\mathbf{x})$ can be readily observed in the *concatenation* property above. However, the real computational advantage comes from the *convolution* property, since it replaces the differential operation of a member in image scale-space by an integration (i.e., convolution) of the generator and an analytical kernel. Using the scale-space notation, the convolution property can be rewritten as

$$\frac{\partial^{i+j}}{\partial x^i \partial y^j} I_\sigma(\mathbf{x}) = \psi_{ij}(x, y; \sigma) * I(\mathbf{x}). \quad (3.7)$$

A scaled image $I_\sigma(\mathbf{x})$ can be represented locally at $\mathbf{x} = (x_1, x_2)$ as ([65]):

$$\begin{aligned} I(x_1 + \xi, x_2 + \eta; \sigma) &= \sum_{n=0}^{\infty} \frac{1}{n!} \left(\xi \frac{\partial}{\partial x_1} + \eta \frac{\partial}{\partial x_2} \right)^n I(\mathbf{x}; \sigma) \\ &= \sum_{n=0}^{\infty} \sum_{i=0}^{n} \binom{n}{i} c_{(n-i)i} \frac{\xi^{n-i} \eta^i}{n!}, \end{aligned} \quad (3.8)$$

where

$$c_{pq} = \frac{\partial^{p+q}}{\partial x_1^p \partial x_2^q} I_\sigma(\mathbf{x}) = I(\mathbf{x}) * \psi_{pq}(\mathbf{x}; \sigma) \tag{3.9}$$

If the local expansion of $I_\sigma(\mathbf{x})$ is truncated at k, the resulting representation is called the *local representation of order* k, and c_{pq} is called *local structure of order* k. Hence, all the local structure of order k in the image $I_\sigma(\mathbf{x})$ can be obtained from the *rf* family $\{\psi_{nm} \mid n, m = 0 \dots k\}$. For example, a local representation of order 2 for an image $I_\sigma(\mathbf{x})$ at scale σ is

$$\begin{aligned} I(x_1 + \xi, x_2 + \eta; \sigma) = I_\sigma(\mathbf{x}) &+ I(\mathbf{x}) * (\psi_{10}(\mathbf{x}; \sigma)\xi + \psi_{01}(\mathbf{x}; \sigma)\eta) \\ &+ \frac{1}{2} I(\mathbf{x}) * \left(\psi_{20}(\mathbf{x}; \sigma)\xi^2 + \psi_{11}(\mathbf{x}; \sigma)\xi\eta + \psi_{02}(\mathbf{x}; \sigma)\eta^2\right) \end{aligned}$$

Koenderink and van Doorn [65] maintained that only ψ_{20}, ψ_{21} and ψ_{30} are relevant in visual information processing. However, we will see shortly that the first-order *rf*, ψ_{10}, also plays an important role in the acquisition of local geometric information about contours, including tangents, curvatures and curvature change rates along the contour.

The formulation above emphasizes the structures at the pixel level, e.g., isophotes, patches or blobs. However, the essence of perception is in the representation of variational information. This emphasis on variational information transforms the representation from the intensity level (point-based) to the contour level (curve-based), and we will need a different formulation at the contour level. The differences between the formulations can be described in terms of the mapping from the spatial domain to the representation domain. At the irradiance level of an image, visual information is conveyed through functions of type $I : R^2 \to R$, while at the contour level the information is embedded in functions of type $c : R^2 \to \{0, 1\}$.

In the next section we present a new low-level representation of images using Gabor kernels, which is also based on Gaussian kernels but is not redundant. The representation is more closely related to natural vision systems because of the similarity of its kernel and the physiology of biological cells.

3.1.3 Image Representation Using Gabor Kernels

The Gabor filter in Eq. (2.5) provides us with an alternative interpretation of Gaussian differentials. In the case of nonuniform differentiation of a two-dimensional Gaussian (e.g., ψ_{20}), the shape of the resulting kernel is actually Gabor-like, and the Gabor form allows us to conveniently interpret the differential of the kernel more meaningfully. For example, Gabor kernels can be interpreted in terms of spatial and Fourier localization, spatial orientation, and scale decomposition in Fourier space [28]. Mathematically, representation using Gabor kernels is a complete representation in the sense of the transformation pair in Eqs. (2.7) and (2.8).

In the Gaussian kernel representation, the information in an image is overlapped in such a way that finer scale representations always contain coarser scale representations (see above). On the other hand, the Gabor kernels represent the information in an image by non-overlapping filters, each of which contribute to the complete representation. From an information-theoretic point of view, a non-redundant representation is clearly desirable, but this representation issue is complicated by the need to compute local variational information efficiently.

3.2 Contour Models

Contours are part of the hierarchy in the geometric structure of images and can be two- or three-dimensional. However, depending on the dimensionality of the contours (i.e., whether they are embedded in an image or on a surface) or the role they play in the analysis (e.g., identifying the boundary of an object in an image or computing surface shape), the representations can take different forms. We will describe three forms of representation and each of them will play different roles in later chapters.

In the following, the term “curve” and “contour” will be used interchangeably since curve is a geometric term, while the embodiment of it in the context of visual perception is contour.

3.2.1 Models in Geometric Space

Given a point P on a 3D space curve, the Serret-Frenet equations relate an orthonormal frame $\{\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{b}}\}$, known as the *Frenet frame*, to two metric variables, κ and τ, known as *curvature* and *torsion*:

$$\begin{aligned} \hat{\mathbf{t}}'(s) &= \kappa \hat{\mathbf{n}}(s) \\ \hat{\mathbf{n}}'(s) &= -\kappa \hat{\mathbf{t}}(s) - \tau \hat{\mathbf{b}} \\ \hat{\mathbf{b}}' &= \tau \hat{\mathbf{n}}. \end{aligned} \tag{3.10}$$

Given a well-defined curve $\mathbf{c}(s)$ parameterized by the natural parameter s (curve length) in two-dimensional Euclidean space, the Taylor expansion of $\mathbf{c}(s)$ is of the form:

$$\mathbf{c}(s_0 + \epsilon) = \sum_{n=0}^{\infty} \frac{\epsilon^n \mathbf{c}^{(n)}(s_0)}{n!}.$$

Using Eq. (3.10) ($\tau = 0$ in this space), the first three terms (up to second-order differentiation of $\mathbf{c}(s)$) of the expansion can be expressed directly in terms of κ, κ', $\mathbf{t}(s)$ and $\mathbf{n}(s)$:

$$\mathbf{c}(s_0 + \epsilon) = \mathbf{c}(s_0) + \left(\epsilon - \frac{\kappa^2 \epsilon^3}{3!}\right) \mathbf{t}(s_0) + \left(\frac{\kappa \epsilon^2}{2} + \frac{\kappa' \epsilon^3}{3!}\right) \mathbf{n}(s_0) + R \tag{3.11}$$

This is the *local canonical form* of the curve $\mathbf{c}$. The implication of this form is that the curve can be decomposed locally into components along the Frenet frame ($\mathbf{t}$, $\mathbf{n}$) and these components, up to a third-order approximation, can be expressed in terms of κ and κ' (derivative of κ with respect to s). As a matter of fact, the *fundamental theorem of the local theory of curves* asserts that $\kappa(s)$ is all we need to specify the curve uniquely (up to a rigid transform) [31].

3.2.2 Models in Signal Space

A piecewise-continuous curve is a one-dimensional geometric object and has infinite bandwidth if treated as a two-dimensional object. By imposing a finite-bandwidth constraint, a cluster of curves can be represented in the two-dimensional domain. It is shown here how this is achieved. In later chapters, it will be used to construct an invertible hyperspace for their representation.

3.2.2.1 2D Curve Representation Using Gaussian Filtering

Consider a planar curve $\mathbf{c} : \mathbf{c}(s)$ in two-dimensional Euclidean space. The two-dimensional representation of $\mathbf{c}(s)$ in the image plane is defined by

$$R_{\mathbf{c}}(\mathbf{x}) \triangleq \int_{-\infty}^{\infty} \delta(\mathbf{x} - \mathbf{c}(s))\, ds \tag{3.12}$$

where $\delta(\mathbf{x})$ is the two-dimensional Dirac delta function. Since δ is singular and does not have finite spatial or frequency content, a finite approximation of the representation can be derived using a Gaussian kernel:

$$\tilde{R}_{\mathbf{c}}(\mathbf{x}) \triangleq g(\mathbf{x}) * R_{\mathbf{c}}(\mathbf{x}) = \int_{-\infty}^{\infty} g(\mathbf{x} - \mathbf{c}(s))\, ds \tag{3.13}$$

where $*$ is the convolution operator and $g(\mathbf{x})$ is the unit-area two-dimensional Gaussian kernel with symmetric σ :

$$g(\mathbf{x}) \triangleq \frac{1}{2\pi\sigma^2} \exp(-\frac{|\mathbf{x}|^2}{2\sigma^2}).$$

Since $g(\mathbf{x})$ is localized in both the spatial and frequency domains, the approximation essentially imposes a finite-bandwidth constraint on the visual processing system.

3.2.2.2 2D Curve Representation in Fourier Space

In the previous section we showed how a planar curve can be represented in a two-dimensional image space. The representation is localized in its information content. In this section we consider how the information can be represented in the discrete domain and with more global scope (as opposed to a local representation).

The sampling theorem tells us that as long as the bandwidth of a signal is limited, the continuous signal can be represented exactly by a set of discrete samples of the signal. This alternative representation becomes an approximation when the signal does not completely vanish above a finite bandwidth or when the number of samples does not reach what is required by the sampling theorem. Here we provide a rationale for using both approximations when converting representations from the continuous to the discrete domain.

Localized representations have exponentially decaying profiles as provided by the Gaussian kernels. This process produces a limited-bandwidth signal for the two-dimensional image space. On the other hand, Gaussian profiles have the same form in both the spatial and Fourier domain, i.e., smooth and exponentially decaying, and each sample in the Fourier domain carries information that globally affects the signal space. These properties allow us to reduce the number of samples drastically without losing essential information while at the same time benefiting the matching process, which will be explained in the next section. Formally, the Fourier transform of the two-dimensional representation $\mathcal{R}_{\mathbf{c}}$ of the planar curve $\mathbf{c}(s)$ is:

$$
\begin{aligned}
R_{\mathcal{F}(\mathbf{c})}(\boldsymbol{\omega}) &= \mathcal{F}(\tilde{R}_{\mathbf{c}}(\mathbf{x})) \\
&\triangleq \iint_{-\infty}^{\infty} g(\mathbf{x}-\mathbf{c}(s)) \exp(-i\boldsymbol{\omega}\cdot\mathbf{x})\, ds\, d\mathbf{x} \\
&= \exp(-\frac{\sigma^2|\boldsymbol{\omega}|^2}{2}) \int_{-\infty}^{\infty} \exp[-i\boldsymbol{\omega}\cdot\mathbf{c}(s)]\, ds \\
&= A(\boldsymbol{\omega}) \exp(i\theta(\boldsymbol{\omega})),
\end{aligned} \tag{3.14}
$$

where $A(\boldsymbol{\omega})$ is the amplitude part and $\theta(\boldsymbol{\omega})$ is the phase part of the representation. Note that these two parts do not correspond to the terms inside and outside the integral since the magnitude of the integral does not equal unity.

It can be observed in Eq. (3.14) that for each point on the curve only the phase encodes information about the curve. The amplitude term is independent of s and only signifies how the energy is constrained locally.

3.2.3 Models in Geometric Feature Space

The geometry of a curve is embedded in the description of the curve, and there is in general no canonical description. However certain descriptions do have parameters that are meaningful for visual perception. These perceptually relevant parameters are *features* of the objects being described and the collection of features along with the characteristics of the geometric space constitute a *geometric feature space*. In other words, a geometric feature space is the result of mapping prominent geometric metrics in the geometric space onto itself. The term "prominent" is a subjective one and is more or less perception-oriented. Here features are taken to be curvature extrema and the curve is derived from the interpolation of feature points as proposed by Attneave [8]. Hermite splines are the primary interpolant used to recover the curve. It is known that piecewise interpolating the feature points on a planar curve can compensate for the oscillatory behavior of a polynomial interpolant, since the order of the polynomial can be minimized. This is especially true when we choose the *knots* of the interpolation to be the highly curved areas, namely, curvature extremum points. Additionally, the availability of higher order differentials at the feature points enable us to proceed without estimating these properties, which is the primary drawback of Hermite interpolation. Hence, a planar curve can be concisely understood from its curvature changes. By identifying curvature extremum points, the curve can be represented by interpolating only the position and first-order derivatives of these points. Hence, the extremum changes of curvature in planar curves are not only important locally but also strongly constrain the global curve shape. This aspect is demonstrated here using Hermite interpolation.

An observer can generally perceive a particular contour at different scales by simply controlling his visual receptors. Hence the structure of a two-dimensional contour varies according to the one-dimensional parameter family of scales [114]. It will be assumed that a proper scale has been chosen for the contour and the concentration will be on second-order differential properties of the contour. It should be noted that by computing in scale space with sampled data, there is really no significant difference computationally between local and global properties.

3.2.3.1 Curvature Extrema as Feature Points

For a continuous and smooth (up to second-order differentials) two-dimensional curve, the structure of the curve is determined by the curvature (Eq. (3.15)) whose extrema define “features” along the curve. A complementary characterization of a curve segment with features is a *featureless* curve segment, which is a curve segment with no curvature extrema between two end points and hence is a region where curvature either remains constant or changes monotonically. We will use the term *local extension* to indicate the procedure to compute the curve segments between feature points (i.e., curvature extrema).

The explicit relationship between a curve and its curvature, parameterized by the curve length is the following. Given the curvature function $\kappa(s)$, the planar curve $\mathbf{c}(s)$ with the specified curvature can be determined up to a translation (a, b) and rotation ϕ as:

$$\mathbf{c}(s) = \left(\int \cos\theta(s)\, ds + a, \int \sin\theta(s)\, ds + b \right) \tag{3.15}$$

where

$$\theta(s) = \int \kappa(s)\, ds + \phi.$$

3.2.3.2 Curve Representation

Since the curvature for the part of curve between adjacent extrema is relatively flat and smooth, the reconstruction problem will be to interpolate the curve using these feature points and their associated geometric properties, i.e., we want to find a curve segment connecting two points $P_1(\mathbf{x}(t_1))$ and $P_2(\mathbf{x}(t_2))$, where both first (tangent) and second (curvature) derivatives are known. Let the kth derivative of P_i with respect parameter t is given by P_i^k. This goal can be accomplished using Hermite interpolation:

$$\mathbf{x}(t) = \sum_{i=1}^{2}\sum_{k=0}^{2} P_i^k H_{k,i}(\frac{t-t_1}{t_2-t_2})$$

where $H_{k,i}$ are Hermite functions (Figures 3.2 and 3.3).

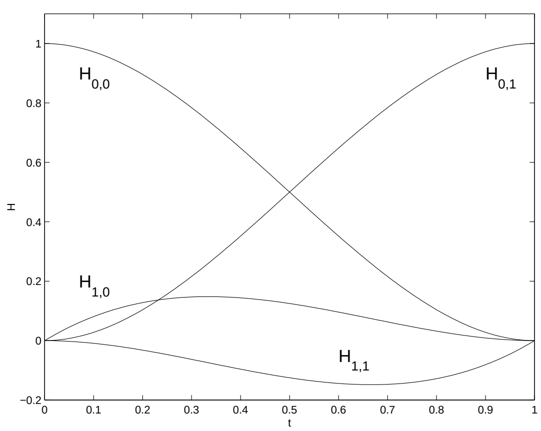


Figure 3.2: Cubic Hermite splines.

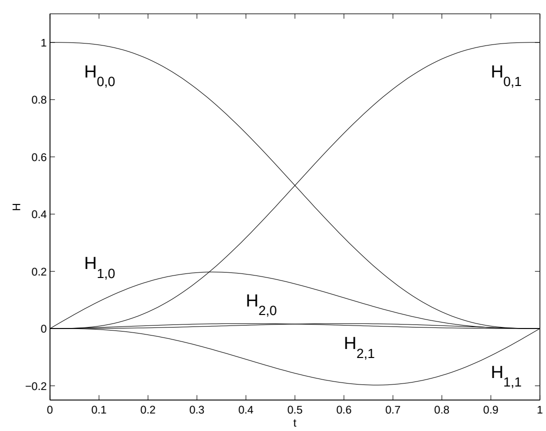


Figure 3.3: Quintic Hermite splines.

We test the formulation in Figure 3.4. The “centripetal” parameterization [70] is used, which takes the square root of the cord length between P_1 and P_2 as the curve parameter t. The curvature versus t plot is given in Figure 3.5 with the corresponding feature points identified in both plots. Using the position coordinates, and the first and second derivatives at the feature points, the interpolated curve is shown in Figure 3.6 and the corresponding curvature is plotted in Figure 3.7, both as solid lines. The original curve and curvature are given as dashed lines.

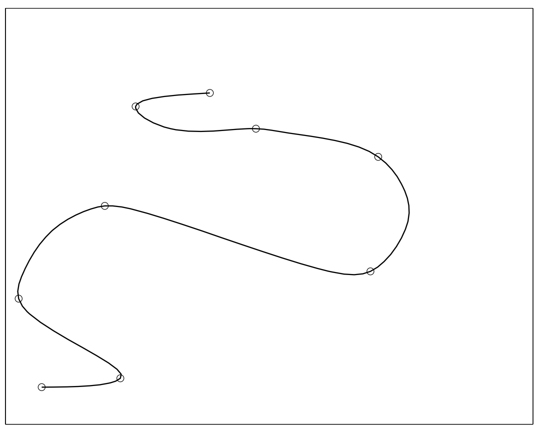


Figure 3.4: A planar curve with feature points identified.

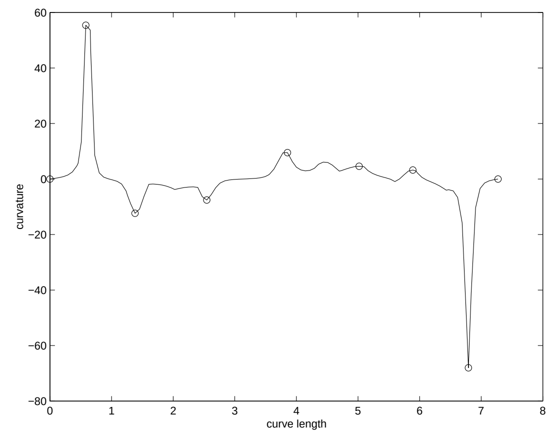


Figure 3.5: Curvature plot of the planar curve.

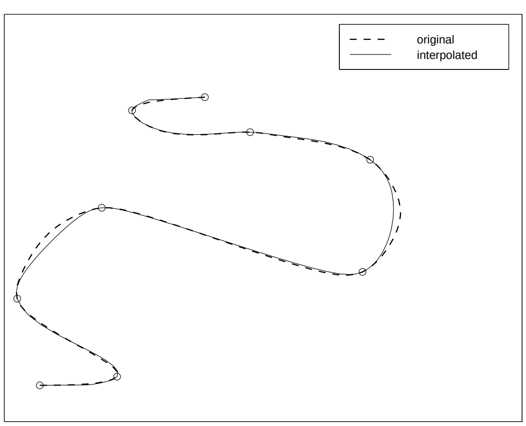


Figure 3.6: The Hermite spline curve using identified feature points.

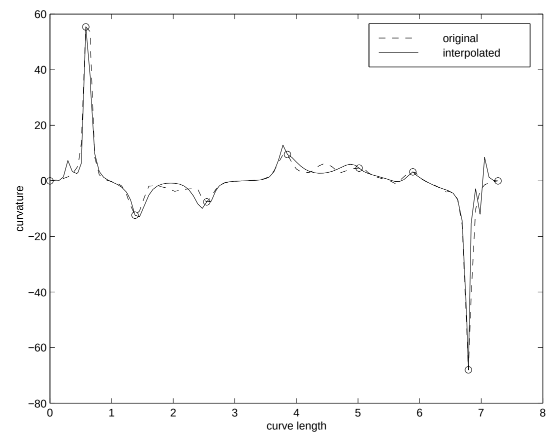


Figure 3.7: Curvature plot of the spline curve with centripetal parameterization.

We also compare a quintic Hermite spline with a cubic Hermite spline, which does not use second derivative information. The curve is shown in Figure 3.8 and the corresponding curvature is shown in Figure 3.9. It can be seen that the cubic splines are as good as the quintic

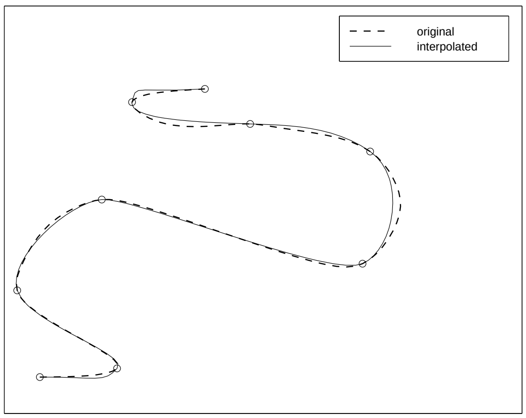


Figure 3.8: Hermite spline curve using only first order derivative.

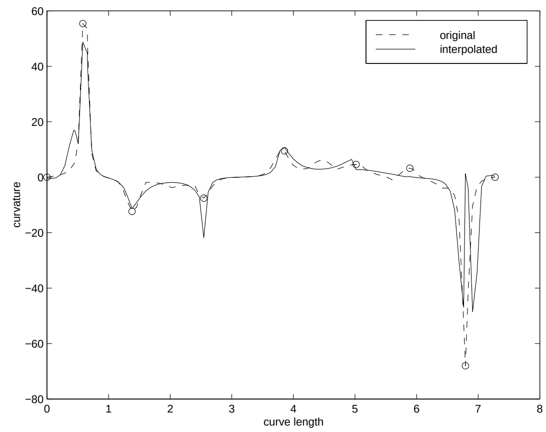


Figure 3.9: Curvature plot of the spline curve with only 1st derivative.

splines, though the latter also interpolate the curvature. This shows the dominant importance of feature locations and their tangent vectors. This is also attributed to the fact that $H_{i,j}$ for $i, j = 0, 1$, show all the essential curvature features, including reflection points.

3.3 Models for Local Surface Shape

The primitives for describing surfaces are intrinsic to the surfaces (intrinsic frame) and independent of the coordinate system (viewer's frame) used to describe them. The lowest order local properties that are intrinsic are differential curvatures. However, perceptually, differential variation is indiscernible and local properties can never be computed reliably. Hence we need a global way to constrain and guide the local computation. Since local computation is what happens (being observed) in the first place, any global (top-down) constraints can be considered as a "guess" and can only be verified at the local level. From the point of view of a mobile observer, global constraints are guides for motion so that object shape can be revealed and represented efficiently.

One of the natural global constraints pertaining to local intrinsic properties is the normal curvature passing through a specific point on the surface. From the theorem on local surface geometry, it is already known that three of these normal curvatures toward different directions determine local shape completely. On the other hand, a single surface curve says something qualitatively about the surface irrespective of how it is projected into the image plane. If the projection can be characterized formally, the curve-surface becomes quantitatively related. One such formal surface curve which does not require observer motion is the apparent contour, since the projected curvature is the normal curvature along the tangent direction of the apparent contour. However, any static surface curve can be so characterized when the observer can move voluntarily (see Chapter 6).

Surface shapes can be acquired from multiple sources. The traditional shape-from-X methods do not require a mobile observer. In the paradigm of active vision, shape-from methods compute surface shape from deformations of computable surface properties, such as image curves [26] and apparent contours [25]. For these curvilinear-based methods, surface shape is computed in a stripe-like way and the whole two-dimensional surface is assembled after enough stripes are acquired.

The combination of the capability of an active observer and the shape cues provided by two-dimensional surface patches rather than curvilinear features will enable us to compute surface shape in a *batch* fashion.

3.3.1 Surface from Triangulated Normal Interpolation

3.3.1.1 Triangulated Patches

Consider any three points, P_1, P_2, P_3 on an object surface that can be tracked by an observer. Assume these points are close enough so that the surface enclosed by them can be considered flat. This assumption can be verified algorithmically later. Since these points are close enough, the projection model can be orthographic since the effect of perspective projection is negligible. Let the length of the segment connecting these points be d_{ij}, where $i, j = 1, 2, 3$ and their corresponding projections onto the image plane be ρ_{ij}. If the observer's frame is oriented toward the surface and the z-axis of the frame is the viewing direction, then foreshortening dictates that

$$\rho_{ij} = d_{ij} \cos\theta$$

where θ is the angle between viewing direction and the surface normal $\hat{\mathbf{n}} = (n_x, n_y, n_z)$ of the plane defined by the three points. If we rotate the viewing direction by an amount of $\delta\theta$, the foreshortening will be

$$\rho'_{ij} = d_{ij} \cos(\theta \pm \delta\theta)$$

where the sign depends on the sign of $\rho_{ij} - \rho'_{ij}$. Solving these two equations we get

$$\tan\theta = \pm \left(\cot\delta\theta - \frac{\rho'_{ij}}{\rho_{ij} \sin\delta\theta} \right).$$

Hence for any single segment between two points, the equation indicates that the surface normal has to lie somewhere on a cone with z-axis being the center axis and the apex angle being θ, i.e., $\hat{\mathbf{n}} = (n_x, n_y, \cos\theta)$, since $\hat{\mathbf{n}} \cdot \hat{\mathbf{z}} = \cos\theta$. If we choose the point $P_1 = (x_1, y_1, z_1)$ as our reference point and compute θ from d_{12}, then since the plane also passes through P_2 and P_3 we have

$$\hat{\mathbf{n}} \cdot \mathbf{d}_{12} = \hat{\mathbf{n}} \cdot \mathbf{d}_{13} = 0$$

i.e.,

$$\begin{pmatrix} n_x & n_y & \cos\theta \end{pmatrix} \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{pmatrix} = \begin{pmatrix} n_x & n_y & \cos\theta \end{pmatrix} \begin{pmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{pmatrix} = 0.$$

Solving for n_x and n_y we have

$$
\begin{aligned}
n_x &= \frac{(z_2 - z_1)y_3 - (z_3 - z_1)y_2}{x_2 y_3 - x_3 y_2} \\
n_y &= \frac{-x_3(z_2 - z_1) + x_2(z_3 - z_1)}{x_2 y_3 - x_3 y_2}
\end{aligned}
\tag{3.16}
$$

Since $(z_j - z_i)/\rho_{ij} = \tan\theta$ we have

$$
\begin{aligned}
n_x &= \frac{\rho_{12} y_3 - \rho_{13} y_2}{x_2 y_3 - x_3 y_2} \tan\theta \\
n_y &= \frac{-\rho_{12} x_3 + \rho_{13} x_2}{x_2 y_3 - x_3 y_2} \tan\theta
\end{aligned}
\tag{3.17}
$$

Let

$$
\mu = \frac{u_x}{u_y} = \frac{\rho_{12} y_3 - \rho_{13} y_2}{-\rho_{12} x_3 + \rho_{13} x_2}
$$

and we have

$$
\hat{\mathbf{n}} = \begin{pmatrix} \dfrac{\mu \sin\theta}{(1 + \mu^2)^{1/2}} & \dfrac{\sin\theta}{(1 + \mu^2)^{1/2}} & \cos\theta \end{pmatrix}. \tag{3.18}
$$

Hence the plane is completely defined.

3.3.1.2 Surface Normal Interpolation

An object surface can be explored by tracking a set of points and triangulating the surface with patches. Since theoretically with an external reference frame this process can be carried out with arbitrary precision, we will consider that the tracked points are data with no measurement errors. On the other hand, we want to restrict our shape computation to be based solely on static features on the surface and in the case of point features, the surface formed by a triangular patch is the best approximation we can get from this approach. Hence the patch normal computed in this way is also considered to be accurate and is essentially a *sampled* representation of the object surface.

The first approximation we are going to make for global surface computation and representation is to determine surface normals at points other than within the patches, and these points are where patches meet each other, that is, the vertices (Figure 3.10).

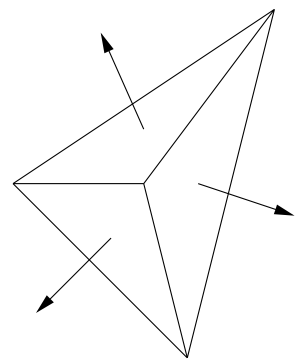


Figure 3.10: Compute vertex normal from neighboring patches.

The real object surface will pass through the vertex and all its neighboring vertices and, among all the possible surfaces, we are only interested in those surfaces that are smooth within each patch and are characterized by a surface normal that is coincident with the patch normal. The best guess we are going to make is the surface that can be fit to the above constraints with the *least-square-error* criterion.

Let the vertex whose normal is to be computed be the origin of our coordinate system and all the other points are expressed in this frame. Since the accuracy of subsequent computation is determined by how we choose our basis, we can make a first approximation of the vertex normal by averaging over the adjacent patches. Let the coordinate basis be $\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$, $\hat{\mathbf{w}}$ and approximating bivariate function be $f(u, v) = au^2 + buv + cv^2 + du + ev$. If there are M neighboring vertices (u_i, v_i, w_i) and N adjacent patches with patch normal $n^j = (n_u^j, n_v^j, n_w^j)$, we can choose a *characteristic* point (u_j, v_j) within each patch in the $u - v$ plane. The normal computed from $f(u, v)$ at $u = u_j$ and $v = v_j$ will be

$$\mathbf{n}^j = \mathbf{x}_u \times \mathbf{x}_v = (-(2au_j + bv_j + d), -(bu_j + 2cv_j + e), 1)$$

where $\mathbf{x}_u = (1, 0, 2au_j + bv_j + d)$ and $\mathbf{x}_v = (0, 1, bu_i + 2cv_j + e)$ are tangent vectors of $f(u, v)$ at (u_j, v_j). The function $f(u, v)$ can be determined by minimizing the least-square-error:

$$\begin{aligned}\epsilon(a, b, c, d, e) = &\sum_{i=0}^{M-1} [w_i - f(u_i, v_i)]^2 + \sum_{j=0}^{N-1} \left[\frac{n_u^j}{n_w^j} + 2au_j + bv_j + d \right]^2 \\ &+ \sum_{j=0}^{N-1} \left[\frac{n_v^j}{n_w^j} + 2cv_j + bu_j + e \right]^2.\end{aligned} \tag{3.19}$$

The solution will be the linear system defined by

$$\frac{\partial \epsilon}{\partial a} = \frac{\partial \epsilon}{\partial b} = \frac{\partial \epsilon}{\partial c} = \frac{\partial \epsilon}{\partial d} = \frac{\partial \epsilon}{\partial e} = 0.$$

Letting $g(u_j, v_j) = n_u^j / n_w^j + 2au_j + bv_j + d$ and $h(u_j, v_j) = n_v^j / n_w^j + 2cv_j + bu_j + e$, the linear equations become

$$\sum_j g(u_j, v_j) = \sum_j h(u_j, v_j) = 0 \tag{3.20}$$

and

$$\begin{aligned}&\sum_i [w_i - f(u_i, v_i)]\, u_i = 0 \\ &\sum_i [w_i - f(u_i, v_i)]\, v_i = 0 \\ &\sum_j \left[g(u_j, v_j) v_j + h(u_j, v_j) u_j \right] - \sum_i [w_i - f(u_i, v_i)]\, u_i v_i = 0.\end{aligned} \tag{3.21}$$

Let $\bar{u} = \frac{1}{N} \sum_j u_j$, $\bar{v} = \frac{1}{N} \sum_j v_j$ and $\bar{\alpha} = \frac{1}{N} \sum_j \frac{n_u^j}{n_w^j}$, $\bar{\beta} = \frac{1}{N} \sum_j \frac{n_v^j}{n_w^j}$. From Eq. (3.20) we have

$$\begin{aligned}d &= -(\bar{\alpha} + 2a\bar{u} + b\bar{v}) \\ e &= -(\bar{\beta} + 2c\bar{v} + b\bar{u}).\end{aligned} \tag{3.22}$$

Substituting d and e into Eq. (3.21) we have

$$\begin{aligned}
\sum_i \left(w_i + \bar{\alpha}u_i + \bar{\beta}v_i\right) u_i &= a\sum_i (u_i - 2\bar{u})\, u_i^2 + b\sum_i (u_i v_i - u_i\bar{v} - v_i\bar{u})\, u_i \\
&\quad + c\sum_i (v_i - 2\bar{v})\, u_i v_i \\
\sum_i \left(w_i + \bar{\alpha}u_i + \bar{\beta}v_i\right) v_i &= a\sum_i (u_i - 2\bar{u})\, u_i v_i + b\sum_i (u_i v_i - u_i\bar{v} - v_i\bar{u})\, v_i \\
&\quad + c\sum_i (v_i - 2\bar{v})\, v_i^2
\end{aligned} \tag{3.23}$$

and

$$\begin{aligned}
& a\left[\sum_i u_i^3 v_i + 2\sum_j u_j v_j - 2\bar{u}\left(\sum_i u_i^2 v_i + N\bar{v}\right)\right] \\
& + b\left[\sum_i u_i^2 v_i^2 + \sum_j \left(u_j^2 + v_j^2\right) - \bar{v}\left(\sum_i u_i^2 v_i + N\bar{v}\right) - \bar{u}\left(\sum_i u_i v_i^2 + N\bar{u}\right)\right] \\
& + c\left[\sum_i u_i^3 v_i + 2\sum_j u_j v_j - 2\bar{v}\left(\sum_i u_i v_i^2 + N\bar{u}\right)\right] = \\
& \sum_i \left[w_i + \bar{\alpha}u_i + \bar{\beta}v_i\right] u_i v_i + N\left(\bar{\alpha}\bar{v} + \bar{\beta}\bar{u}\right) - \sum_j \left[\frac{n_u^j}{n_w^j} v_j + \frac{n_v^j}{n_w^j} u_j\right].
\end{aligned} \tag{3.24}$$

Solving for a, b, and c using Eqs. (3.23) and (3.24), we can compute $\hat{\mathbf{n}}$ at the vertex from Eq. (3.22) as

$$\hat{\mathbf{n}} = \frac{(-d, -e, 1)}{(1 + d^2 + e^2)^{1/2}}.$$

3.3.2 Surface Curvatures

There are several ways to compute surface curvatures once the surface normal is determined for a set of points on the surface. In this section we introduce two methods and analyze their accuracy and error propagation behavior.

3.3.2.1 Euler's Formula

Given at least three patches around a given vertex on an object surface, Euler's formula (i.e., the second fundamental form in the local frame defined by the principal directions) can be used to determine the principal directions and principal curvatures.

Given three normal curvatures, $\kappa_1, \kappa_2, \kappa_3$, Euler's formula relates them to the two principal curvatures, $\kappa_{\max}, \kappa_{\min}$, by the equations:

$$\begin{aligned}
\kappa_1 &= \kappa_{\max} \cos^2 \theta + \kappa_{\min} \sin^2 \theta \\
\kappa_2 &= \kappa_{\max} \cos^2(\theta + \phi_2) + \kappa_{\min} \sin^2(\theta + \phi_2) \\
\kappa_3 &= \kappa_{\max} \cos^2(\theta + \phi_3) + \kappa_{\min} \sin^2(\theta + \phi_3),
\end{aligned} \tag{3.25}$$

where θ is the angle between the direction of κ_1 and the maximum principal curvatures, and ϕ_2, ϕ_3 are the angles between κ_2, κ_1 and κ_3, κ_1, respectively (see Figure 3.11). Solving for $\theta, \kappa_1, \kappa_2$ we have

$$\theta = \frac{1}{2} \tan^{-1} \left(2 \frac{\kappa_1(\cos^2 \phi_3 - \cos^2 \phi_2) + \kappa_2(1 - \cos^2 \phi_3) + \kappa_3(\cos^2 \phi_2 - 1)}{\kappa_1(\sin 2\phi_2 - \sin 2\phi_3) + \kappa_2 \sin 2\phi_3 - \kappa_3 \sin 2\phi_2} \right) \tag{3.26}$$

and

$$\begin{aligned}
\kappa_{\max} &= \frac{\kappa_2 \sin^2 \theta - \kappa_1 \sin^2(\theta + \phi_2)}{\cos^2(\theta + \phi_2) - \cos^2 \theta} \\
\kappa_{\min} &= \frac{\kappa_1 \cos^2(\theta + \phi_2) - \kappa_2 \cos^2 \theta}{\cos^2(\theta + \phi_2) - \cos^2 \theta}.
\end{aligned} \tag{3.27}$$

In this method any error in computing the surface normal will propagate to the curvature computation and will be amplified by the differential process used in computing normal curvature along the associated patch directions. One way to improve on this is not to compute the normal curvatures and subsequently solve Euler's formula but to use the normal interpolated at the vertex and fit a bivariate function on the tangent plane at the vertex. This method will be

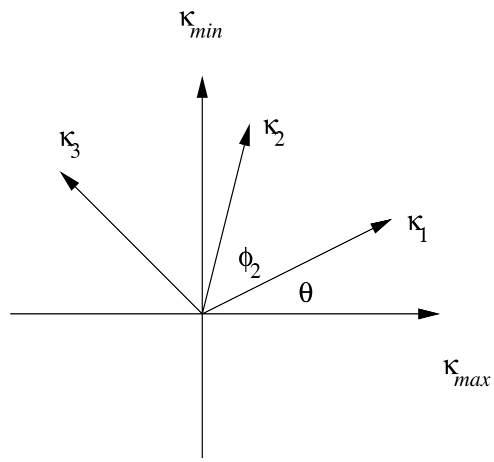


Figure 3.11: Computing the principal curvatures using Euler's formula.

discussed next.

3.3.2.2 Bivariate Approximation

Given a vertex, the surface normal at this point and all its neighboring points, we can fit a bivariate function on the tangent plane defined by the surface normal [44]. This is the function of lowest order that has the same tangent plane and curvature as the object surface at the given point.

Given the normal $\hat{\mathbf{n}}$ at the point, we can express all coordinates and vectors in the tangent frame (Figure 3.12) ($\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$, $\hat{\mathbf{n}}$) with $\hat{\mathbf{n}} = \hat{\mathbf{u}} \times \hat{\mathbf{v}}$ and the bivariate function will have the form:

$$f(u, v) = au^2 + buv + cv^2.$$

If we rotate the orthogonal ($\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$) frame around $\hat{\mathbf{n}}$ by an angle θ the new representation of

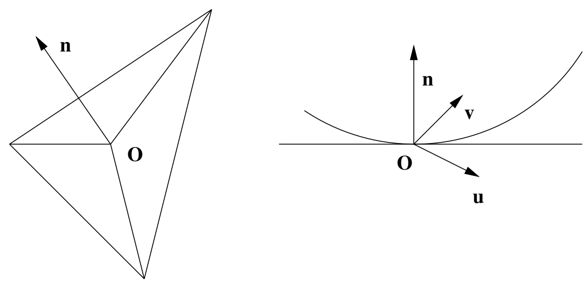


Figure 3.12: Compute curvature using bivariate interpolation.

$f(u, v)$ will be

$$\begin{aligned} g(u', v') &= f(u' \cos\theta - v' \sin\theta, u' \sin\theta + v' \cos\theta) \\ &= \left(a \cos^2\theta + b \sin\theta \cos\theta + c \sin^2\theta\right) u'^2 + ((c-a)\sin 2\theta + b \cos 2\theta)\, u'v' \\ &+ \left(a \sin^2\theta - b \sin\theta \cos\theta + c \cos^2\theta\right) v'^2. \end{aligned} \tag{3.28}$$

By choosing

$$\theta = \frac{1}{2} \tan^{-1} \frac{b}{a-c}$$

we can eliminate the cross term $u'v'$ and identify the principal curvatures as

$$\begin{aligned} \kappa_1 &= a \cos^2\theta + b \sin\theta \cos\theta + c \sin^2\theta \\ \kappa_2 &= a \sin^2\theta - b \sin\theta \cos\theta + c \cos^2\theta \end{aligned} \tag{3.29}$$

For a set of neighboring vertices $P_i, i = 0 \dots m-1$, the desired equation is

$$\begin{pmatrix} u_0^2 & u_0 v_0 & v_0^2 \\ \vdots & \vdots & \vdots \\ u_{m-1}^2 & u_{m-1} v_{m-1} & v_{m-1}^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} w_0 \\ \vdots \\ w_{m-1} \end{pmatrix} \triangleq \mathbf{Ax} = \mathbf{w}.$$

Solving the least-square-error equation is equivalent to solving the *normal equation*

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{w}.$$

However, normal equations are generally *ill-conditioned* and tend to be singular or susceptible to rounding errors. To remedy this, we use, instead, the *singular value decomposition* method to solve the problem. The matrix **A** has a singular value decomposition of the form

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

where **S** is a diagonal matrix with its diagonal elements (s_0, s_1, s_2) being the *singular values* of the matrix **A**. It can be proved that the vector

$$\mathbf{x} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{w}$$

solves the least-square-error problem. The ill-conditioned difficulty can be resolved by setting the inverse of small singular values to zero.

The accuracy of this method will be governed by how the surface normal is estimated, and one way to estimate the surface normal at a given point is to solve the normal interpolation problem and the resulting bivariate function.

3.4 Summary

The mathematical foundation for handling geometric structures in images was presented in this chapter. The emphasis was on the computation and representation of the information models for a computational vision system. Correspondence between perceptual features and geometric models of curves and surfaces was established. These relationships as well as their computational formulation will be used extensively in the remainder of this thesis.

Formal components in the models have been interpreted in the context of visual perception as well. The logical question of how may these geometric features be computed in the local computation model (Chapter 2) is answered in the next chapter.

Chapter 4

2D Local Curve Computation

Images as sources of visual information consist of organized variations of brightness on the image plane. The front-end of the system is designed so as to select only those parts that have spatial variations (see Chapter 2). In addition to this selection process, the hypothesis that the image is a representation of the physical world directs the system to search for spatial organization that is relevant and organized. The organization considered here is geometric (see Chapter 3), and the language used is differential calculus, which is justified from the scale-space formulation of images and their representations.

In this chapter the correspondence between perception and geometric features in 2D domain is investigated by formulating the computational mechanism that is responsible for computing these perceptual features. It is shown that not only the contours (commonly referred as "edges" when considered not structured) but also the tangents, curvatures and higher-order intrinsic invariants can be computed from the mechanism of receptive fields, which are local computationally and, henceforth, can be considered part of the front-end. This result makes formal models such as the local canonical form of a curve particularly meaningful.

The chapter begins by formulating the contour computation in the framework of antisymmetric receptive fields. Computing tangents is a logical development of this formulation. This is followed by showing how curvature and derivative of curvature can be computed in the same framework. Various examples are given to demonstrate the effectiveness of the computational procedure. The chapter concludes with some considerations on the relationship between local geometric features and visual attention control.

4.1 Contour and Its Geometric Invariants

The *convolution property* (Eq. (3.5)) provides us with a means to replace differential operations with integral operations—a process that averages out noise rather than amplifies it (see Section 4.3.2 for some discussion on differentiation using the convolution property). This property, combined with the language of scale space and differential geometry, enables us to systematically compute some of the most useful geometric invariants of contours.

4.1.1 Contour

A curvilinear contour is locally linear and is characterized locally by its tangent vector $\hat{\mathbf{t}} = (\cos\theta, \sin\theta)$. An operator that can signal the presence of a local linear segment must have a preferred response in the direction θ. One of the simplest prototypical filters in our family of receptive fields is ψ_{01} (Figure 4.1), which has preferred orientation in the $\hat{\mathbf{y}}$ direction. The filter form of arbitrary orientation can be acquired through rotation of coordinate systems.

When a coordinate system is rotated by an angle θ, the coordinate of a point (x, y) in the original system is now represented by a new coordinate (x^R, y^R) given by

$$(x^R, y^R) = (x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta). \tag{4.1}$$

Define a 2-D antisymmetric receptive field with orientation θ as [66]:

$$P_k(x, y, \theta; \sigma) \overset{\triangle}{=} -\psi_{01}(x^R, y^R; \sigma) \tag{4.2}$$

A local image contour with orientation θ at scale σ is defined to be a distribution of irradiance $I(x, y)$ such that

$$
\begin{aligned}
&[(\nabla P_k(x, y, \theta; \sigma) \cdot \mathbf{n}) * I(x, y)] = 0, \quad \text{and} \\
&[P_k(x, y, \theta; \sigma) * I(x, y)] \neq 0
\end{aligned}
\tag{4.3}
$$

where $\hat{\mathbf{n}} = (\sin\theta, -\cos\theta)$ is the normal vector to the contour (i.e., $\hat{\mathbf{t}}$). The term $(\nabla P_k \cdot \mathbf{n})$ is the directional derivative of $P_k(x, y, \theta; \sigma)$ in the direction of $\hat{\mathbf{n}}$ and, in terms of the receptive field ψ (see Eq. (3.3) and Figure 3.1), has the explicit form:

$$
\nabla P_k \cdot \hat{\mathbf{n}} = \psi_{02}(x^R, y^R; \sigma). \tag{4.4}
$$

The response of the image to the kernel P_k has maximum rate of change when moving in the direction orthogonal to $\hat{\mathbf{t}}$ (see Figure 4.1). The additional condition is needed to exclude uniform contrast areas of the image. Note that $\psi_{02}(x^R, y^R; \sigma)$ is in the form of a Gabor filter (Eq. (2.5)). The location (x, y) defined by Eq. (4.3) is the maximum response of the antisymmetric kernel P_k along the direction $\mathbf{n}$ and is analogous to the output of an oriented edge detector using the Gabor kernel.

In the following we will drop the σ term in various expressions when it is clear that we are dealing with a particular scale σ.

4.1.2 Tangential Field Along a Contour

The tangent vector at (x, y) and scale σ along an image contour (Eq. (4.3)) is defined as the unit vector $\hat{\mathbf{t}}$ with orientation θ such that

$$
\Phi(x, y, \theta; \sigma) \triangleq \frac{\partial P_k(x, y, \theta; \sigma)}{\partial \theta} * I(x, y) = 0 \tag{4.5}
$$

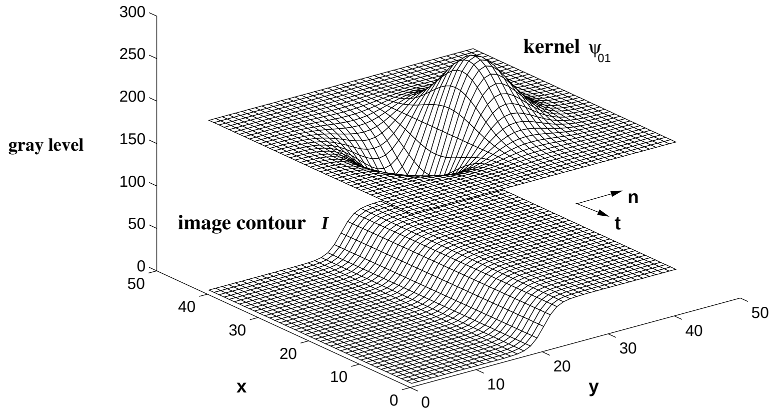


Figure 4.1: Contour defined by the response of image to ψ_{01} kernel at a particular orientation.

This definition comes from the fact that when kernel $P_k(x, y, \theta)$ is aligned with the local image contour (with orientation θ), the response of convolving the kernel with the image will be maximum with respect to the orientation parameter θ. The equation $\Phi(x, y, \theta; \sigma) = 0$ implicitly defines θ as a function of (x, y), that is, $\theta = \theta(x, y)$.

Equation 4.5 is in the form of convolving a kernel with the image $I(x, y)$. Hence it is convenient to consider properties of the kernel alone and call the kernel "associated" with the resulting function after the convolution operation. Define the kernel associated with $\Phi(x, y, \theta)$ as

$$\phi(x, y, \theta) \stackrel{\triangle}{=} \frac{\partial P_k(x, y, \theta)}{\partial \theta} = \psi_{10}(x^R, y^R) \tag{4.6}$$

where (x^R, y^R) is given by Eq. (4.1). For a given point (x, y), the *orientation space* at this point is defined as

$$\Psi(\theta; \sigma) = \phi(x, y, \theta; \sigma) * I(x, y).$$

4.1.2.1 Estimation of Tangent

Since θ is a continuous parameter, the orientation of the tangent can only be estimated by quantizing the orientation space, i.e., we need to determine the resolution of the orientation space in order to locate zero points accurately.

Physiological evidence suggests a quantization resolution of $\pi/18$ (36 quantizations) for the mammalian vision system [53]. It is shown in this section that a resolution of $\pi/4$ (8 quantizations) is sufficient if we assume a step edge model.

The orientation space at the origin for a horizontal step edge is given by

$$\Psi(\theta;\sigma) = \int_{-\infty}^{0}\int_{-\infty}^{\infty} \phi(x,y,\theta;\sigma)\,dx\,dy = \frac{\sin\theta}{\sqrt{2\pi}\,\sigma} \tag{4.7}$$

where the edge is going from 1 to 0 when crossing from the negative y-axis to the positive y-axis. The above expression is the output of applying the local kernel $\phi(x, y, \theta)$ to the step edge image. We would like to find the θ that defines the tangent field without knowledge of the closed-form solution (which is $\sin\theta$ for a step edge but unknown otherwise).

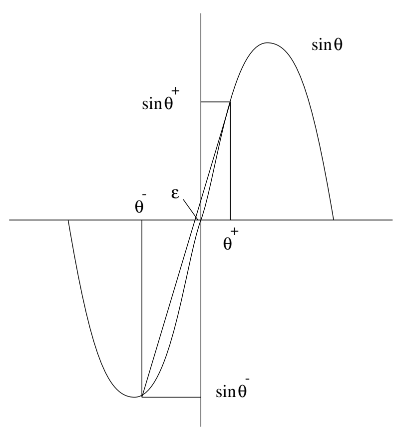


Figure 4.2: The response of a step edge to the ϕ kernel. The ideal response is the sinusoid. The zero-crossing point of the sinusoid can be approximated by a straight line connecting a point on the negative and a point on the positive side.

Since the sinusoidal function is linear around the zero point, we can estimate the resolution needed by estimating the linearity of the sinusoidal function in the range $(-\pi, \pi)$. If we denote the two points around 0 as θ^- and θ^+ for negative and positive orientation samples, respectively, then the error between the linear approximation and the actual sinusoid will be (see Figure 4.2)

$$\epsilon = \frac{\theta^- \sin\theta^+ - \theta^+ \sin\theta^-}{\sin\theta^+ - \sin\theta^-}. \tag{4.8}$$

Hence, by keeping θ^- and θ^+ within $\pi/4$ we can keep the estimated error of the zero point within 1.3° for this case. The estimated zero point $\theta(x, y)$ is

$$\theta(x, y) = \frac{\Phi(x, y, \theta^+)\theta^- - \Phi(x, y, \theta^-)\theta^+}{\Phi(x, y, \theta^+) - \Phi(x, y, \theta^-)}. \tag{4.9}$$

4.1.3 Curvature Along a Contour

By definition, the curvature κ is $d\theta/ds$, where s is the natural parameter (curve length). For a given tangent field, $\theta(x, y)$, using the chain rule and implicit differentiation, we have

$$\kappa = \frac{\partial\theta}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial\theta}{\partial y}\frac{\partial y}{\partial s} = \nabla\theta \cdot (\cos\theta, \sin\theta). \tag{4.10}$$

The explicit form of $\nabla\theta$ can be acquired by differentiation of the equation $\Phi(x, y, \theta) = 0$ (defined in Eq. (4.5) with respect to x and y, since, for example,

$$\frac{\partial\Phi(x, y, \theta(x, y))}{\partial x} = \frac{\partial\Phi(x, y, \theta)}{\partial x} + \frac{\partial\Phi(x, y, \theta)}{\partial\theta}\frac{\partial\theta}{\partial x} = 0$$

and it is straightforward to show that

$$\nabla\theta = -\frac{\nabla\Phi}{\Phi_\theta}, \tag{4.11}$$

where $\Phi_\theta \stackrel{\triangle}{=} \partial\Phi/\partial\theta$. Hence

$$\kappa = -\frac{\nabla\Phi \cdot \mathbf{t}}{\Phi_\theta} \tag{4.12}$$

For simplicity we will use subscripts to denote derivatives (e.g., Φ_θ for $\partial\Phi/\partial\theta$). As stated above, since $\Phi(x, y, \theta)$ is defined as $\phi(x, y, \theta) * I(x, y)$, we can directly associate Φ with the kernel ϕ. Using the same terminology, the explicit forms of kernels $\phi_x, \phi_y, \phi_\theta$ associated with Φ_x, Φ_y and Φ_θ, respectively, are:

$$\begin{aligned}
\phi_x(x, y, \theta) &= \psi_{20}(x, y)\cos\theta + \psi_{11}(x, y)\sin\theta \\
\phi_y(x, y, \theta) &= \psi_{02}(x, y)\sin\theta + \psi_{11}(x, y)\cos\theta \\
\phi_\theta(x, y, \theta) &= -\psi_{10}(x, y)\sin\theta + \psi_{01}(x, y)\cos\theta = \psi_{20}(x^R, y^R).
\end{aligned} \tag{4.13}$$

The explicit expressions for $\nabla\Phi \cdot \mathbf{t}$ and $\nabla\Phi \cdot \mathbf{n}$ are

$$\nabla\Phi \cdot \mathbf{t} = \psi_{20}(x^R, y^R) * I(x, y) \tag{4.14}$$

$$\nabla\Phi \cdot \mathbf{n} = \psi_{11}(x^R, y^R) * I(x, y) \tag{4.15}$$

It should be noted that these two expressions are invariant with respect to rotation. From Eq. (4.14) and Eqs. (4.12) and (4.13) we have the explicit form of curvature at (x_0, y_0), which can be directly used for computation:

$$\kappa(x_0, y_0) = \frac{\iint_{-\infty}^{\infty} \psi_{20}(x^R, y^R) I(x_0 - x, y_0 - y)\, dx\, dy}{\iint_{-\infty}^{\infty} \psi_{01}(x^R, y^R) I(x_0 - x, y_0 - y)\, dx\, dy} \tag{4.16}$$

A similar formulation of curvature was proposed by Koenderink *et al.* [61] for image blob boundaries defined by iso-luminance (the neighborhood around a point on an image contour can be approximated by an iso-luminance contour). However, the tangent orientation cannot be computed accurately in their formulation and their expression of curvature for image contours can only be approximated by a third-order differential.

4.1.4 Derivative of Curvature Along a Contour

The same method used to derive curvature can also be applied to formulate higher-order geometric invariants. In particular, we will derive the derivative of curvature, $d\kappa/ds$, since it is also part of the expression of the local canonical form and possesses perceptual importance [48]. Using Eq. (4.10) the differentiation of curvature with respect to curve length is

$$\begin{aligned}\frac{d\kappa}{ds} &= \theta_{xx}\cos^2\theta + \theta_{yy}\sin^2\theta + 2\theta_{xy}\sin\theta\cos\theta \\ &\quad - (\theta_x\sin\theta - \theta_y\cos\theta)\kappa\end{aligned} \tag{4.17}$$

We have already derived $(\theta_x, \theta_y) = -(\Phi_x, \Phi_y)/\Phi_\theta$. By differentiating this equation with respect to x and y, we can derive θ_{xx}, θ_{yy} and θ_{xy} in terms of various orders of differentiation of Φ:

$$\begin{aligned}\theta_{xx} &= -(\Phi_{xx}\Phi_\theta^2 - 2\Phi_x\Phi_\theta\Phi_{x\theta})/\Phi_\theta^3 \\ \theta_{yy} &= -(\Phi_{yy}\Phi_\theta^2 - 2\Phi_y\Phi_\theta\Phi_{y\theta})/\Phi_\theta^3 \\ \theta_{xy} &= -(\Phi_{xy}\Phi_\theta^2 - \Phi_x\Phi_\theta\Phi_{y\theta} - \Phi_y\Phi_\theta\Phi_{x\theta})/\Phi_\theta^3\end{aligned} \tag{4.18}$$

Hence we have expressed the derivative of curvature in terms of first and second order differentiation of $\Phi(x, y, \theta)$. Using Eq. (4.13) we can directly compute ϕ_{xx}, ϕ_{xy} and ϕ_{yy}. If we define

$$\lambda \triangleq -\frac{\nabla\Phi \cdot \mathbf{n}}{\Phi_\theta} \tag{4.19}$$

then it can be shown that

$$\frac{d\kappa}{ds} = \kappa\lambda - \frac{1}{\Phi_\theta}\left(\psi_{30}(x^R, y^R) * I(x, y)\right). \tag{4.20}$$

This formula can also be derived by applying the directional derivative of κ in the $\mathbf{t}$ direction.

4.2 Examples

The images in Figure 4.3 will be used to illustrate the method. Initially the scale-orientation

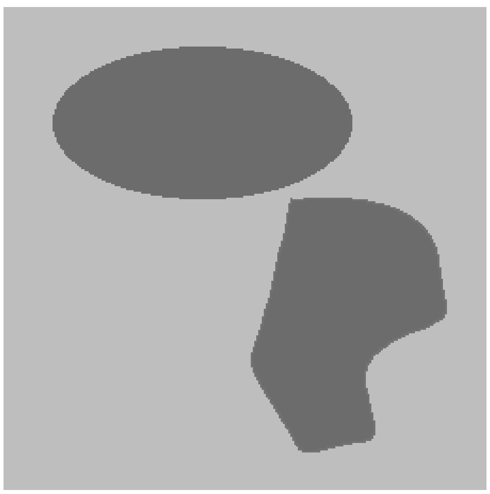


Figure 4.3: Image of two synthetic geometric shapes.

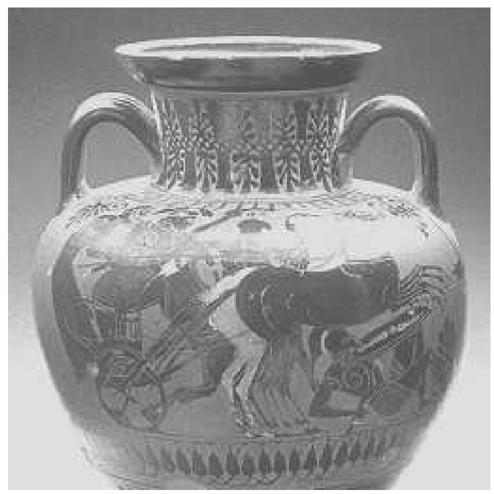


Figure 4.4: Image of a vase from Smithsonian archive.

space is partitioned into 4×4 cells, i.e., using four scale partitions of $\sigma = 1.5, 2, 3, 4$ and four orientation partitions of $\theta = 0, \pi/2, \pi, 3\pi/2$. This scheme enables us to locate contours through operations in the Fourier domain, which is equivalent to performing operations uniformly in the image domain. At this stage θ is treated as a quantized parameter and does not get estimated. Next, the orientation space is repartitioned into eight cells and the tangent field is estimated along the contours. In this second pass θ is treated as a continuous parameter, and the computation is conducted in the image domain at those contour points.

For the ellipse in Figure 4.3, the theoretical and estimated tangent are shown in Figure 4.5. The orientation is plotted versus the curve length along the ellipse. Similar comparison is also done for curvature and is shown in Figure 4.6. The computation of the curvature and derivative of curvature for this image are shown in Figure 4.7 and Figure 4.8 respectively. The accuracy and correctness of the results can be best observed by examining them along the boundary of the irregular shape.

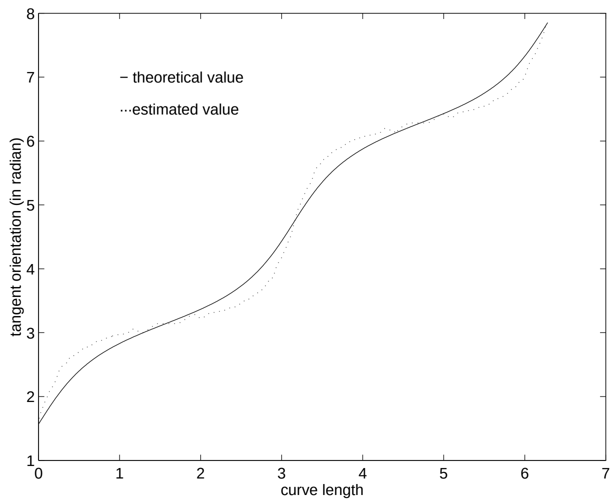


Figure 4.5: Comparison between theoretical and computed tangent along the boundary of the ellipse in Figure 4.3.

We also computed the curvature for the vase image along the left boundary and the top (an ellipse) of the vase and the result is shown in Figure 4.9. The highest peak of the curvature comes from the concave discontinuity near the vase handle.

4.3 Discussion

We have shown that for each of the invariants in the local canonical form of a contour, we can derive a set of local kernels that can be used to compute the invariant directly from the raw image. The steps are: (1) compute image contours using the kernel $(\nabla P_k \cdot \mathbf{n})$ (Eq. (4.3)), (2) compute the vector tangent fields for $I(x, y)$ and express them in the form of (x, y, θ),

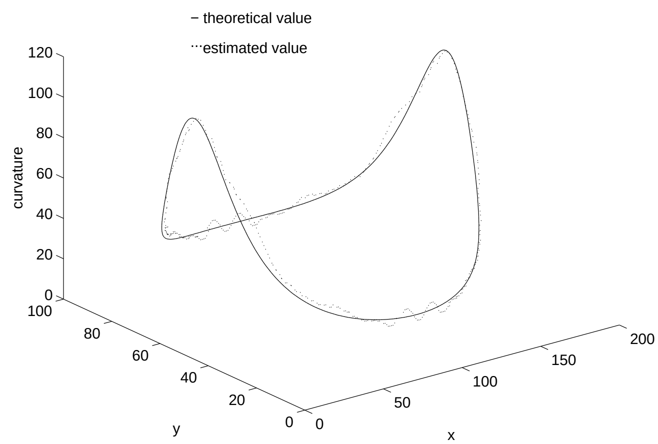


Figure 4.6: Comparison between theoretical and estimated curvature along an ellipse.

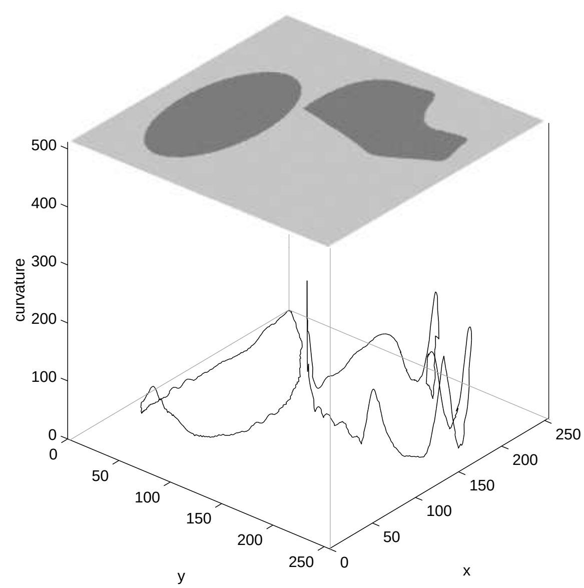


Figure 4.7: Computed curvature for Figure 4.3.

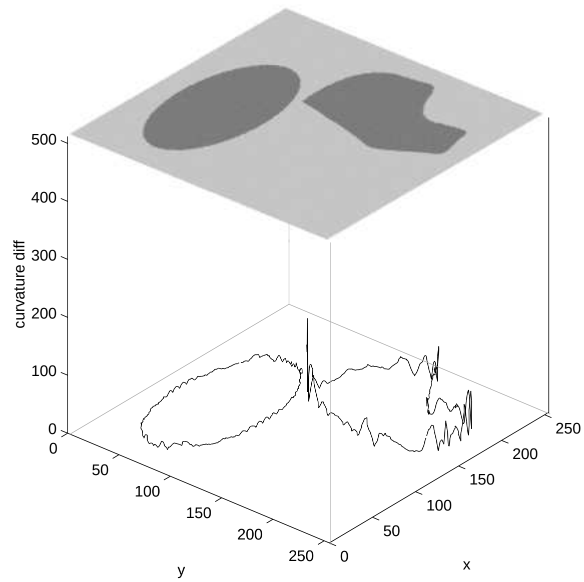


Figure 4.8: Curvature derivative of Figure 4.3.

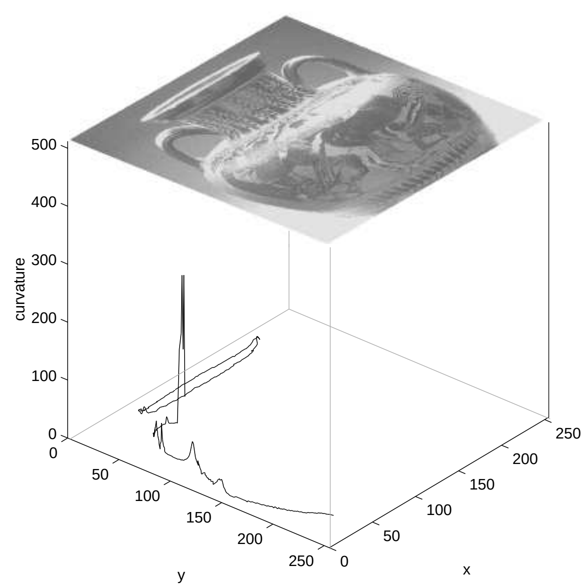


Figure 4.9: Curvature computation of the left boundary of the vase image.

where the vector field $\mathbf{t}$ is $(\cos\theta, \sin\theta)$ (Eq. (4.5)), and (3) for points where the tangent field is non-vanishing, compute the curvature (Eq. (4.16)) and, after that, the derivative of curvature (Eq. (4.20)).

The kernels derived above for computing local geometric invariants along image contours can all be found in biological systems [53, 66] though the antisymmetric kernels are not as populated as the symmetric ones. However, they can all be derived from the Gaussian kernel with expanded kernel sizes as the differential order increases. These properties suggest possible connections between the computation of geometric information and the organization of natural visual systems.

In the rest of this section some further considerations about the computation in this approach are discussed.

4.3.1 Scale and Size of Kernels

When a continuous signal is considered in the modeling process, scales are bounded only by the object systems being modeled. However, when the signal is converted to the digital domain by a sampling process, the range of scales is also dictated by the sampling process. In addition, the local kernels of the receptive fields will increase in size due to this conversion.

The expressions for $\psi_i(x; \sigma)$ in Eq. (3.4) are all normalized so that the total area under each is unity. This is important since they function as filters on images. In order to maintain this property, the normalization factor is proportional to the order of differentiation. This implies an expansion of the filter size for a constant numerical precision. This increase of the kernel size also constrains the range of scales because the bandwidth of the sampled image is constrained by the Nyquist rate. In fact, the image size determines both the upper and lower bounds of the scale space. If the scale is taken to be multiples of σ (e.g., scale = $\alpha\sigma$, with $\alpha \in I$) in the Gaussian kernel, then the upper bound is $\sigma_{\max} = N/2\alpha$, where the image size is $N \times N$. On the other hand, taking N as the Nyquist rate dictates the scale lower bound to be $\sigma_{\min} = \alpha/\pi$.

4.3.2 Differentiation Using Convolution Property

Replacing differentiation by integration using the convolution property is advantageous under two conditions: (1) the available information is considered valid only above a certain scale σ_0, and (2) there is no truncation in computing the convolution. Because the Gaussian filter does not preserve the function it operates on (for images, it is the image generator), the differentiation obtained by the convolution property is not the real differentiation of the function but a smoothed version of it. The second condition is necessary to avoid truncation noise. Weiss [112] proposed to replace the Gaussian filter with *power preserving filters* for these two reasons. The first condition is equivalent to treating fine variations below σ_0 as sampling noise, which should be discarded by the implied smoothing of the receptive field $\psi_{ij}(\mathbf{x}; \sigma)$ for $\sigma > \sigma_0$. In a biological system, these fine variations are smoothed out by the sampling mosaic and the temporal response characteristics of each receptive field. The second condition is satisfied when $\sigma_{\min} < \sigma < \sigma_{\max}$.

4.3.3 Contour and Tangent Computations

Theoretically, we can use Eq. (4.3) to compute the tangent orientation θ. However, as described earlier, we can compute image contours in the Fourier domain by treating θ as a quantized parameter. This greatly increases the efficiency of computation at the price of being less precise in estimating θ. On the other hand, after potential contours are located, we need only compute the geometric properties at these contour points and, because of the sinusoidal property indicated by Eq. (4.7), we can then estimate θ with great precision.

4.3.4 Curvature and Foveation

Curvature has been considered as an informative feature for foveation—a process of shifting visual attention to a specific part of the image and conducting detailed analysis of the local region

[8, 32]. This process is illustrated in Figure 4.10 and 4.12, in which curvature extrema are used to locate foveation points for detailed contour analysis. Figure 4.13 illustrates the contours that have connected paths to the attended points, while Figure 4.14 shows the magnitude-encoded curvature along these contours.

Figure 4.10: Attention points for the vase image.

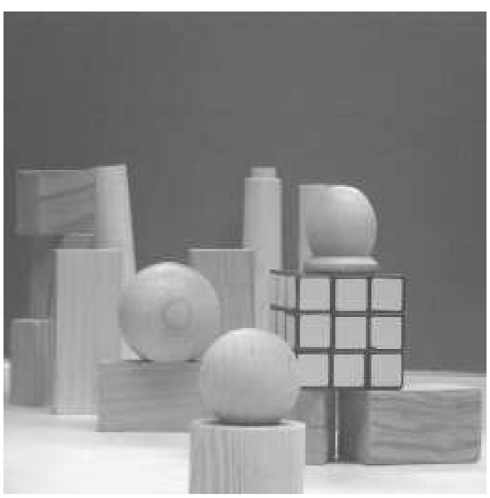


Figure 4.11: An image of miscellaneous shape of blocks.

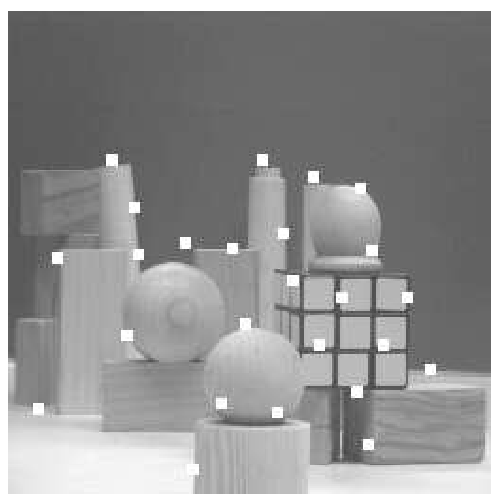


Figure 4.12: Attention points for the block image.

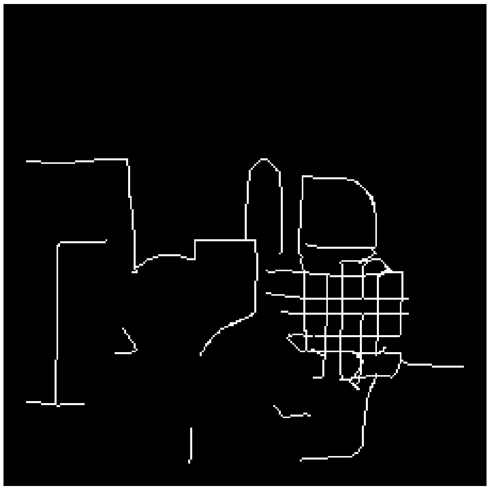


Figure 4.13: Spatial localization using attention points.

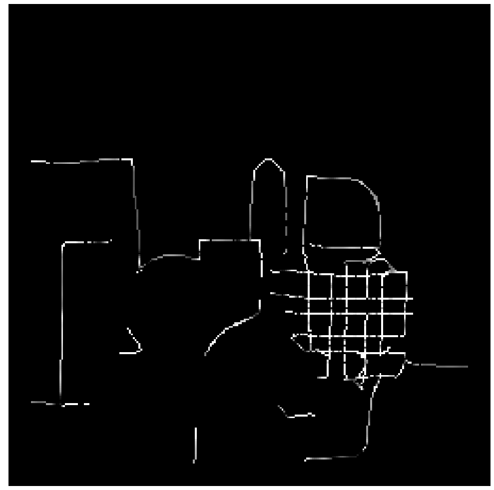


Figure 4.14: Curvature along contours with magnitude encoding.

4.4 Summary

In this chapter it was shown that various local geometric invariants of image contours can be computed directly and reliably from an image as part of the local computation process (Figure 4.15). This new approach eliminates the drawbacks of error propagation in conventional approaches and the need for global processes such as energy minimization.

Being able to accurately and reliably compute higher order differential invariants such as derivative of curvature allows us to explore the connection between perception and these geometric invariants (e.g., curve partitioning and representation [48]). This also makes two-dimensional visual processes such as perceptual organization more meaningful.

The ability to compute local geometric features at the early stages of processing provides us with a much more powerful set of primitives to work with. This is essential for geometry in 3D domain and global representation to be disccussed in the following chapters.

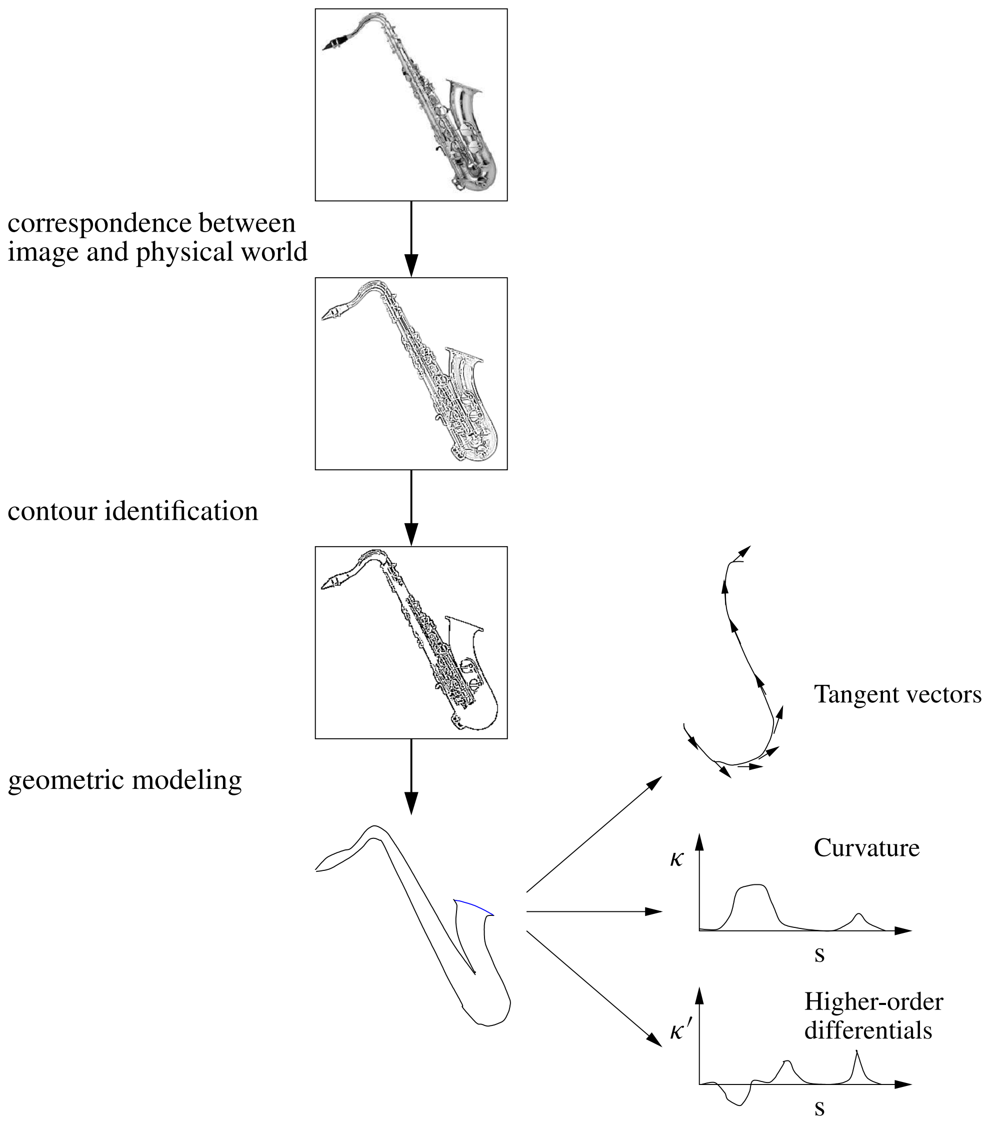


Figure 4.15: The process of computing geometric information from an image.

Chapter 5

Global 2D Curve Description

Perception is global phenomenon. However, the foundation of a vision system is its local computational process. Therefore it is necessary to bridge the gap between local computations and global results that characterize visual perception. In mathematical analysis, global attributes of functions are the results of integration on local properties, which are analyzed by differentiation. For example, the property that a surface is curved can only be established if we observe two adjacent geodesics on the surface for a substantial distance. If the deviation of the two geodesics varies from a constant, then the surface is curved. This approach cannot be used in perception simply because the vision system is not a good analyzer This is exactly why the earth appears to be flat to us, and it is even harder for us to accept that the physical 3D space of our existence could be curved. In contrast, the vision system relies on strategic identification of local features in space and tries to generalize the local shape of space into a global characterization. This happens in both 2D and 3D spaces.

One specific question is the focus of this chapter: what kinds of global processes that are meaningful for visual perception can be defined on a set of local properties? This is not restricted to elements of a particular geometric language. In essence, it is part of an attempt to capture the capability of an observer who can identify 2D patterns that do not have an explicit organization. This is exemplified by the recognition of patterns such as a seemingly random combination of curves without being able to describe its geometric structure. Hence, the local elements used here are not geometric but point-based. In other words, the underlying space is the signal space rather than the information space.

In this chapter a Fourier-based representation scheme for 2D objects is developed that preserves both curve information and is stable against input noise. A high dimensional representation space is derived from this scheme and it is shown that, for each type of viewpoint-dependent variation, there is a corresponding well-defined matching process which is independent of the size of the database to be matched against. The sensitivity of the matching processes under various variations is also analyzed. Examples of object recognition from a database of musical instruments are shown. The algorithm is also shown to be effective in the presence of a combination of perspective, translation, scaling, and rotation transformations.

This chapter is organized as follows. The mathematical construct of the abstract space that contains each curve as a subspace (that is, hyperplane) is described first. This is followed by showing that the set of all affine transformations (i.e., a rigid transformation plus scaling) for an object can be formulated using this hyperplane geometry and the matching process has constant time complexity. Examples are given at the end of the chapter.

5.1 Representation Space $\mathcal{D}$

The Fourier representation fully preserves the spatial information in a curve—one of the important criteria in curve representation. However, the representation itself is not invariant to several important transformations. This deficiency can be remedied by introducing a new space spanned by the Fourier coefficients and studying the subspace induced by these transformations.

The 2D representation $\mathcal{R}_{\mathbf{c}}$ of a planar curve $\mathbf{c}(s)$ is given by Eq. (3.12). Its Fourier transform is given by Eq. (3.14). Because of the localization property of $R_{\mathcal{F}(\mathbf{c})}(\boldsymbol{\omega})$, there is a neighborhood with radius μ around the origin in Fourier space where a given percent of the energy is within a given bound, i.e., μ is considered to be the approximation of the bandwidth of $\tilde{R}_{\mathbf{c}}(\mathbf{x})$. Under this approximation, the required number of samples along a single dimension will be 2μ and for the 2D representation of $\mathbf{c}(s)$ the total number of samples will be $4\mu^2$ when using a rectangular sampling grid. Let the samples be at $\boldsymbol{\omega}_{ij} = (\omega_i, \omega_j)$, where $i, j \in [0, n-1]$ (hence, $n = 2\mu$), and the coefficients of $R_{\mathcal{F}(\mathbf{c})}$ at $\boldsymbol{\omega}_{ij}$ be c_{ij} (a complex number in either Cartesian form or polar form). Consider a one-to-one mapping from the pair (i, j) to an integer: $f(i, j) = k$. Let the inverse mapping be $(i, j) = (h_1(k), h_2(k))$. Using the mapping $f(i, j)$, c_{ij} can be re-indexed as c_k. Denote the polar form of c_k as (r_{2k}, r_{2k+1}), that is,

$$c_{ij} = R_{\mathcal{F}(\mathbf{c})}(\boldsymbol{\omega}_{ij}) = c_{h_1(k)h_2(k)} \stackrel{\triangle}{=} c_k = r_{2k} \exp(i r_{2k+1}) \tag{5.1}$$

Consider the tuples:

$$\mathbf{r} = (r_0, r_1, \cdots, r_{N-1}) \tag{5.2}$$

where $N = 8\mu^2$ and the new Euclidean space $\mathcal{D}$ contains points $\mathbf{r}$. For each $k = 0, \ldots, N-1$, designate a point along the kth dimension in $\mathcal{D}$ with coordinate r_k. Under this construction, the curve $\mathbf{c}$, as represented in Fourier space, becomes a point in $\mathcal{D}$ with coordinates $\mathbf{r} = (r_0, \ldots, r_{N-1})$ (Figure 5.1). The capacity of space $\mathcal{D}$ depends on the resolution and range of the Fourier coefficients. If the coefficients are all converted to integer and are in the range $[0, M]$, the total capacity of $\mathcal{D}$ is M^N.

Next, some critical properties regarding stability, rigid transformation, and scaling of $R_{\mathcal{F}(\mathbf{c})}$ in space $\mathcal{D}$ will be analyzed.

5.1.1 Stability of Representation in $\mathcal{D}$

Traditional methods for curve representation generally involve thresholding when the curve is converted from piecewise-continuous to either piecewise-linear or a selected feature space.

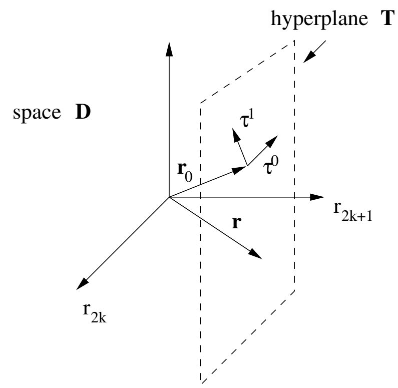


Figure 5.1: Representation space $\mathcal{D}$ and hyperplane $\mathcal{T}$ spanned by $\boldsymbol{\tau}^0$ and $\boldsymbol{\tau}^1$.

Whenever thresholding is involved, the representation will not be stable against input noise. In another words, given a measure of distance, a new representation due to the finite variation of a curve at a certain point does not have a bounded distance from the original representation. Since no such thresholding process is involved when converting the curve $\mathbf{c}(s)$ to the $N-$dimensional representation space $\mathcal{D}$, and all the mappings are continuous and invertible, the point $\mathbf{r}$ in $\mathcal{D}$ is stable with respect to local perturbation of $\mathbf{c}(s)$.

5.1.2 Translation in $\mathcal{D}$

Let the $\mathcal{D}$ space representation of a curve $\mathbf{c}(s)$ be $\mathbf{r}_0 = (r_0, \ldots, r_{N-1})$. A translated curve of $\mathbf{c}(s)$ along a vector $\mathbf{p}$ has the form $\mathbf{c}_t(s) = \mathbf{c}(s) - \mathbf{p}$. The corresponding Fourier representation is

$$R_{\mathcal{F}(\mathbf{c}_t)}(\boldsymbol{\omega}) = \exp(i\boldsymbol{\omega} \cdot \mathbf{p})R_{\mathcal{F}(\mathbf{c})}(\boldsymbol{\omega}) = A(\boldsymbol{\omega})\exp[i(\theta(\boldsymbol{\omega}) + \boldsymbol{\omega} \cdot \mathbf{p})] \tag{5.3}$$

Let $R_{\mathcal{F}(\mathbf{c})}(\boldsymbol{\omega})$ be represented by $\mathbf{r}$ in $\mathcal{D}$. It can be observed from Eq. (5.3) that $R_{\mathcal{F}(\mathbf{c-p})}$ has the same amplitude coordinates (those r_{2k}) as $R_{\mathcal{F}(\mathbf{c})}$, while the phase coordinates (r_{2k+1}) at $\boldsymbol{\omega}_{ij}$

undergo a translation (in $\mathcal{D}$) of $-\boldsymbol{\omega}_{ij} \cdot \mathbf{p}$. Hence the translated curve has the representation:

$$\begin{aligned} \mathbf{r} &= \mathbf{r}_0 - (0, \omega_{h_1(0)} p_1 + \omega_{h_2(0)} p_2, \dots, 0, \omega_{h_1(N-1)} p_1 + \omega_{h_2(N-1)} p_2) \\ &\stackrel{\triangle}{=} \mathbf{r}_0 - p_1 \boldsymbol{\tau}^1 - p_2 \boldsymbol{\tau}^2 \end{aligned} \tag{5.4}$$

where $\boldsymbol{\tau}^k = (0, \omega_{h_k(0)}, \dots, 0, \omega_{h_k(N-1)})$ for $k = 1, 2$. In the above we make use of the one-to-one mapping of $(i, j) = (h_1(k), h_2(k))$. The vectors $\boldsymbol{\tau}^1$ and $\boldsymbol{\tau}^2$ are constant vectors determined by the one-to-one mapping functions and Eq. (5.4) defines a parameterized hyperplane $\mathcal{T}$ in $\mathcal{D}$ (Figure 5.1). The hyperplane $\mathcal{T}$ is actually parallel to those amplitude axes defined by $(1, 0, \dots, 0), \dots, (0, \dots, 1, 0)$, i.e., those $N/2$ vectors in $\mathcal{D}$ with 1 at one of its even-indexed coordinates and zero elsewhere. Hence all possible translations of a curve are confined within the hyperplane $\mathcal{T}$ in $\mathcal{D}$. The scope of $\mathcal{T}$ is determined by the scope of possible translation vectors $\mathbf{p}$. We will discuss this aspect when we consider matching in Section 5.2.

5.1.3 Scaling in $\mathcal{D}$

It can be observed that, from Eq. (3.14), the scaled curve $\mathbf{c}_s(s) = a\mathbf{c}(s)$ of $\mathbf{c}(s)$ has the Fourier representation $R_{\mathcal{F}(\mathbf{c}_s)}(\boldsymbol{\omega})$:

$$\begin{aligned} \iint_{-\infty}^{\infty} G(\mathbf{x} - a\mathbf{c}(s)) \exp(-i\boldsymbol{\omega} \cdot \mathbf{x})\, ds\, d\mathbf{x} &= \exp(-\frac{\sigma^2 |\boldsymbol{\omega}|^2}{2}) \int_{-\infty}^{\infty} \exp(-ia\boldsymbol{\omega} \cdot \mathbf{c}(s))\, ds \\ &= \exp\left[-\frac{\sigma^2 |\boldsymbol{\omega}|^2 (1 - a^2)}{2}\right] R_{\mathcal{F}(\mathbf{c})}(a\boldsymbol{\omega}) \end{aligned} \tag{5.5}$$

Thus scaling in the spatial domain has a corresponding scaling effect in the Fourier domain. In order to analyze how this will affect the point $\mathbf{r}$ (representing $\mathbf{c}(s)$), we need to relate the Fourier coefficients at $\boldsymbol{\omega}_{ij}$ with those coefficients at $a\boldsymbol{\omega}_{ij}$. However, there is no direct correlation between these two sets of coefficients and other strategies are called for.

Since $R_{\mathcal{F}(\mathbf{c})}(\boldsymbol{\omega})$ is a continuous function of $\boldsymbol{\omega}$ and is a result of integrating a continuous function of the curve $\mathbf{c}(s)$, it follows that the mappings from $\mathbf{c}(s)$ to $R_{\mathcal{F}(\mathbf{c})}$ to $\mathbf{r}$ in $\mathcal{D}$ are all continuous and smooth. In addition, $\mathbf{r}$ is a continuous function of the scale factor a. Hence, instead of having a scale invariant formulation of matching in $\mathcal{D}$, the Fourier representation $R_{\mathcal{F}(\mathbf{c})}(\boldsymbol{\omega})$ could be scaled in advance and matched in $\mathcal{D}$. This corresponds to the multi-channel model in Chapter 2, where its biological foundation is described.

If we represent a scaled curve in logarithmic space, its Fourier representation embodies an effect identical to the translation case and, for a fixed scale factor, the differential of $R_{\mathcal{F}(\mathbf{c}_s)}(\boldsymbol{\omega})$ with respect to a will be linear in the sum of the frequencies with respect to the Fourier coefficients. That is, in logarithmic space,

$$\frac{\partial R_{\mathcal{F}(\mathbf{c}_s)}(\boldsymbol{\omega})}{\partial a} = \frac{\omega^1 + \omega^2}{a} R_{\mathcal{F}(\mathbf{c}_s)} \tag{5.6}$$

where $\boldsymbol{\omega} = (\omega^1, \omega^2)$. Hence it is natural to divide Fourier space logarithmically during the "pre-scaling" process. This strategy is similar to some used in modeling biological vision systems [76].

5.1.4 Rotation in $\mathcal{D}$

The rotation of $\mathbf{c}(s)$ about the origin is given by

$$\mathbf{c}_r(s)) \triangleq \mathbf{c}(s) \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} = \mathbf{c}(s)\mathbf{Q} \tag{5.7}$$

From Eq. (3.14), $\mathbf{c}_r(s)$ has the Fourier representation of $R_{\mathcal{F}(\mathbf{c}_r)}(\boldsymbol{\omega})$:

$$\begin{aligned} &\iint_{-\infty}^{\infty} G\left(\mathbf{x} - \mathbf{c}(s)\mathbf{Q}\right) \exp(-i\boldsymbol{\omega}\cdot\mathbf{x})\, ds\, d\mathbf{x} \\ &= \exp(-\frac{\sigma^2|\boldsymbol{\omega}|^2}{2}) \int_{-\infty}^{\infty} \exp\left(-i(\boldsymbol{\omega}\mathbf{Q}^T)\cdot\mathbf{c}(s)\right)\, ds \\ &= R_{\mathcal{F}(\mathbf{c})}(\boldsymbol{\omega}\mathbf{Q}^T) \end{aligned} \tag{5.8}$$

since $|\boldsymbol{\omega}| = |\boldsymbol{\omega}\mathbf{Q}^T|$. This is similar to scaling where changes of axes are necessary in $\mathcal{D}$ because the new Fourier coefficients are identical to the original coefficients at a different frequency. Again the solution is to rotate $R_{\mathcal{F}(\mathbf{c}_r)}(\boldsymbol{\omega})$ in the Fourier domain and then convert to the space $\mathcal{D}$ representation for matching. However, since full rotation invariance is generally not implemented by biological systems (for good reasons), we only need to "pre-rotate" $R_{\mathcal{F}(\mathbf{c})}(\boldsymbol{\omega})$ within a range, $[-\phi_r,\ \phi_r]$, where rotation invariance is desired.

5.2 2D Matching

Given the framework presented so far, matching involves locating the point in the space $\mathcal{D}$ representing the curve to be matched. Since there is no representation for curves that is invariant to the effects of translation, scaling and rotation, we need to apply the theory developed thus far regarding these effects.

The matching scheme begins with the specification of a set of 2D planar curves, $\mathbf{c}(s)$, for the characterization of a 2D object. The Fourier representation of each curve is computed using Eq. (3.14) and the corresponding $\mathcal{D}$ space representation $\mathbf{r}_c$ is computed from Eq. (5.1) using the mapping $f(i, j)$ and $h_1(k), h_2(k)$ for each sampling frequency $\boldsymbol{\omega}_{ij} = (\omega_i, \omega_j)$. Matching succeeds if there are points $\mathbf{r}$ in the ϵ-neighborhood of $\mathbf{r}_c$, i.e.,

$$|\mathbf{r} - \mathbf{r}_c| < \epsilon. \tag{5.9}$$

Only at this final decision is a threshold used. At this point the metric in $\mathcal{D}$ and the threshold value ϵ remain to be decided and the decision is independent of the rest of the recognition process.

5.2.1 Translation

The requirement for translation invariance is equivalent to searching a subspace (the hyperplane $\mathcal{T}$) in the representation space $\mathcal{D}$. The scope of the search is governed by the 2D translation vector $\mathbf{p}$. The theoretical values for $\mathbf{p}$ are $([-\infty,\ \infty], [-\infty,\ \infty])$. However, when implemented in a biological system, the visual resolution is not uniform across photo receptors and the visual field is not a full panorama. A general strategy for limiting the values of $\mathbf{p}$ is to bring $\mathbf{c}(s)$ to the center of the visual field (foveation). In this case, $\mathbf{p}$ has scope $([-P,\ P], [-P,\ P])$ and the hyperplane $\mathcal{T}$ as parameterized by Eq. (5.4) has a domain of $2P \times 2P$.

5.2.2 Scaling and Rotation

The control parameter in scaling is the scale factor a which, again, has the theoretical value of $(0, \infty]$. The same argument used in the previous section regarding translation applies and the factor will be big enough for $\mathbf{c}_s = a\mathbf{c}$ to be above the finest visual resolution and small enough for $\mathbf{c}_s$ to be entirely within the visual field. This puts both a lower and an upper limit on the possible values of a, i.e., $a \in [a_{\min}, a_{\max}]$.

In the case of rotation, a linear division between the scope $[-\phi_r,\ \phi_r]$ is needed and the number of divisions will depend on the threshold value ϵ.

5.2.3 Matching Complexity and Partial Matching

It is highly desirable to have a matching mechanism that is approximately independent of the size of the database, at least for a certain plausible size. On the other hand, it is not possible to have a “canonical” representation of an object other than the simplest one. This implies that an object may have multiple presence in the database and this aspect should not have a noticeable impact on matching performance.

Since none of the algorithms described here depends on the size of database, the recognition process has a constant time complexity. Because of this property, it will be beneficial to have multiple representations of a single object, not because of the necessity to "pre-scale" and "pre-rotate" but to encode functional "parts" of an object (Figure 5.2). This redundancy is valuable for recognition under partial occlusion.

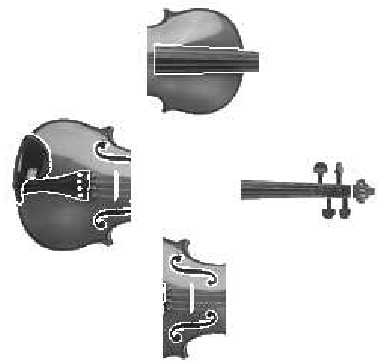


Figure 5.2: Parts of a violin.

5.2.4 Algorithm Summary

Given descriptions of objects in terms of their curvilinear features (Eq. (3.12)), a 2D Fourier representation is acquired using Eq. (3.13) and Eq. (3.14). Based on the resolution used (μ), a subset of the coefficients in the representation is used to construct the representation space $\mathcal{D}$ using Eq. (5.1) and Eq. (5.2). If a similar object is in the database, any translated version of the object will be represented by points within a hyperplane $\mathcal{T}$ in $\mathcal{D}$ (Eq. (5.4)), which has a finite scope of $([-P,\ P], [-P,\ P])$, where P depends on the scope of the optical receptors of the system. The scaled and rotated version of the same object will be represented by unpredictable points in $\mathcal{D}$ (Eq. (5.5) and Eq. (5.8)). This problem can be solved by pre-scaling and pre-rotating the unknown object and matching each one to the database (each matching still has constant time complexity). The final decision of similarity is controlled by the metric in Eq. (5.9).

5.3 Examples

We used a database of musical instruments (Figure 5.3) to test our method. The curvilinear features that characterize each of the objects are shown in Figure 5.4.

Figure 5.3: A database of musical instruments.

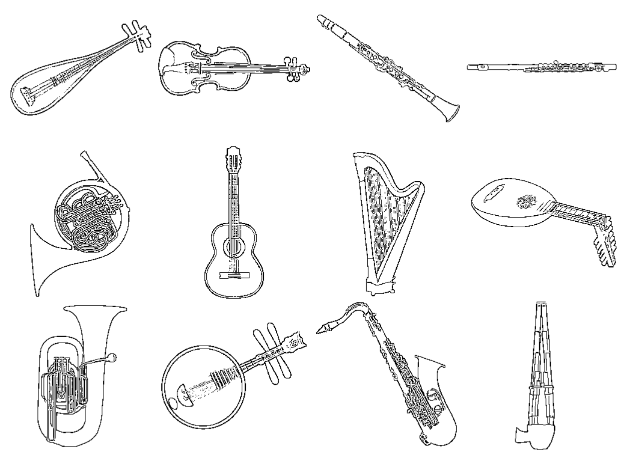


Figure 5.4: Curvilinear features of the database.

Three different resolutions at $\mu = 8, 16, 32$ were used for comparison purposes. The dimensions of $\mathcal{D}$ are 512, 2,048 and 8,192, respectively. The corresponding visual information represented for a double-bass is shown in Figure 5.5. The translation scope P is set to 20 pixels and hence the domain of the hyperplane $\mathcal{T}$ is 40×40. Pre-scaling is done at scale factors 0.75, 0.875, 1.125, and 1.25, while pre-rotation is done at angles from $-\pi/4$ to $\pi/4$ in increment of $\pi/12$.

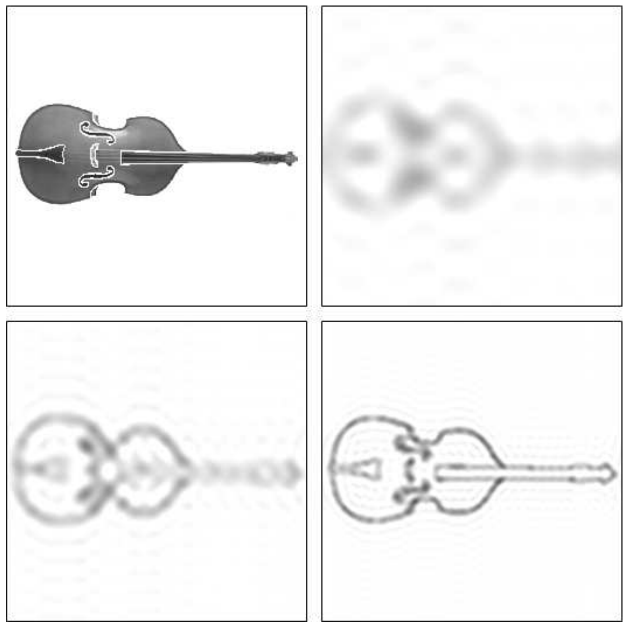


Figure 5.5: Encoding at multiple resolutions ($\mu = 8, 16, 32$) **of a double-bass.**

Different choices of the metric $|\mathbf{r} - \mathbf{r}_c|$ may cause drastically different responses to various transformations. Our choice of the Euclidean metric is because of its smoothness and simplicity. However, this does require moving back and forth between the polar representation used in $\mathcal{D}$ and the Cartesian representation. All matching measures were normalized to the same number of samples.

One of the important parameters is the threshold ϵ for measuring matching similarity. This value depends not only on the amount of curvilinear information of objects being encoded, but also on the sensitivity of the transformations when applied to objects. For a typical object, the sensitivity of distance measures in $\mathcal{D}$ with respect to translation is shown in Figure 5.6, while the sensitivity for scaling and rotation is shown in Figure 5.7 and Figure 5.8, respectively. It can be observed that higher resolution corresponds to higher selectivity against transformations

μ	pipa	violin	clarinet	flute	guitar
8	104475	19365	102446	102856	113172
16	39797	16764	37400	37059	42208
32	16959	11446	16203	15737	17709

Table 5.1: The matching measures of a double-bass against other instruments at various resolutions.

and hence should have a higher value of ϵ for recognizing slightly scaled and rotated objects. For $\mu = 8$ a translation offset around $(\delta x^2 + \delta y^2)^{1/2} = 10$ has about the same error distance as a rotation of $\phi \approx \pi/20$ (i.e., 9°) or as a scale factor of about 0.86. At $\mu = 32$, the same value will only allow $\phi \approx \pi/90$ (i.e., 2°) of rotation and a scale factor of 0.97. On the other hand, at higher resolutions, due to the fact that more complex curvilinear features are compared, ϵ should be lowered in order to recognize the similarity between, say, a double-bass and a violin. In our implementation, the values $\epsilon = 20000$ at $\mu = 8$ and $\epsilon = 12000$ at $\mu = 32$ were chosen because of these considerations.

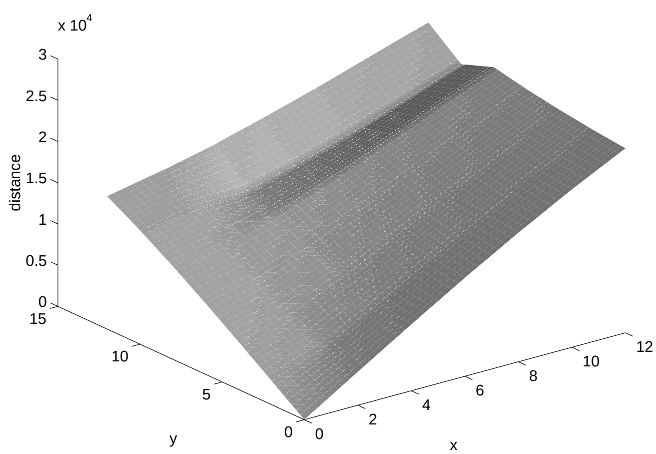


Figure 5.6: The error surface in $\mathcal{D}$ with respect to translation for $\mu = 8$.

The matching measure for the double-bass in Figure 5.5 compared to the database in Figure 5.3 is shown in Table 5.1.

An image of a violin and its curvilinear features under perspective transformation, spatial translation ($(\delta x, \delta y) = (-6, 2)$), rotation ($\phi = \pi/6$), and scaling (factor = 0.898) is shown in

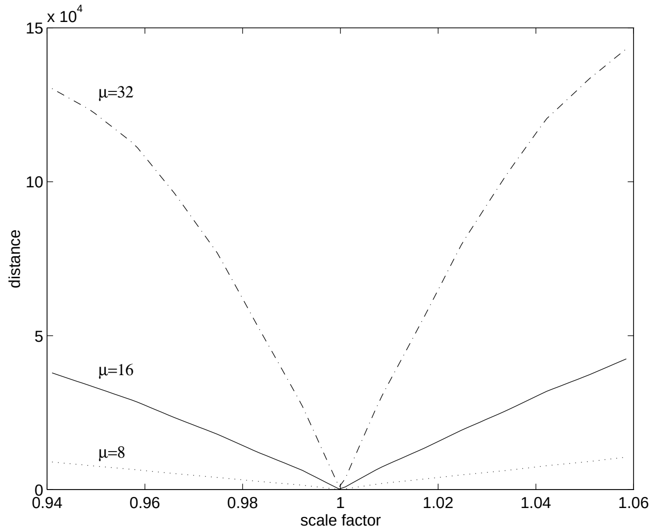


Figure 5.7: Error curves for scaling in $\mathcal{D}$ for $\mu = 8$, 16, 32.

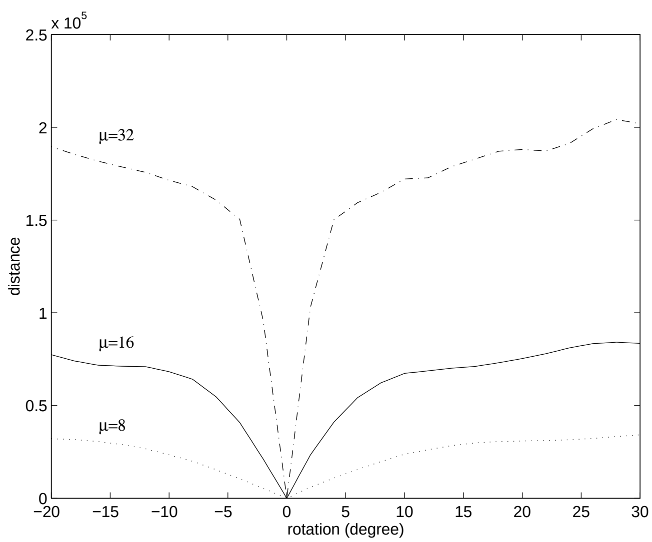


Figure 5.8: Error curves for rotation in $\mathcal{D}$ for $\mu = 8$, 16, 32.

Figure 5.9. The recovered object is shown in Figure 5.10.

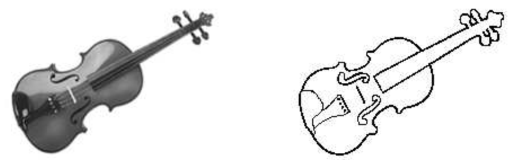


Figure 5.9: An object under various transformations.

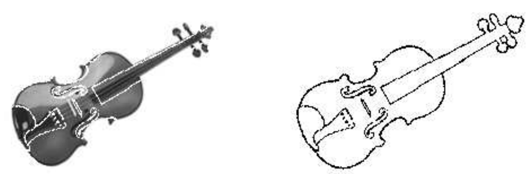


Figure 5.10: Recognized object under the transformation: $(\delta x, \delta y) = (-10, 2)$**,** $\phi = \pi/6$ **and scaling factor=0.875.**

5.4 Discussion

The matching process introduced here is essentially applied directly to the images of curvilinear features. However, the decision regarding which part will constitute the characterizing features is beyond the system described here. In the example, the characterizing features were manually selected. Since the formulations are based on stable curvilinear features of 2D objects, it is highly desirable that only characteristic parts are represented (e.g., see Figure 5.9).

Two interrelated parameters are critical in the system: the resolution μ used in the representation and the similarity criterion ϵ in $\mathcal{D}$. As can be observed in Figure 5.5 and Table 5.1, a higher resolution results in a lower discernible similarity measure, in spite of the visual similarity between a violin and a double-bass. This can be accounted for by noting the fact that more details are being compared. The relationship between these two parameters can further be seen in Figures 5.6, 5.7, and 5.8. The higher the resolution, the higher the sensitivity against variations. In other words, it is harder for the visual system to achieve invariance against view point variations when attending to the details. However, we don't have to compromise by selecting a single resolution representation, since the matching process is independent of the size of the database. As mentioned above, multiple and redundant representations of objects can be valuable in this regard.

The problem of translation invariance has been handled by methods such as using the magnitude part of the Fourier transform [111]. This is equivalent to using the metric (see Eq. (3.14))

$$|\mathbf{r} - \mathbf{r}_c| \stackrel{\triangle}{=} |A(\boldsymbol{\omega} - A_c(\boldsymbol{\omega}|.$$

Since the representation used for measuring similarity does not preserve essential visual information, there is a high-collision rate in the representation space.

The large capacity of the representation space $\mathcal{D}$ implies the sparse nature of the space. This is handled traditionally by *hashing* the representation in $\mathcal{D}$. However, this is a problem at the implementation level.

5.5 Summary

In order for the representation about objects to be useful for recognition as well as other vision tasks, the representation has to be stable against noise while still preserving essential visual information of objects. It is also essential for the recognition system to be invariant to certain classes of variations. This latter goal can be achieved by either designing representations out of desirable invariants or designing a matching process that has a well-defined and predictable behavior in the presence of these variations. In addition, the matching process should be independent of the size of the database in order to handle real-world applications. In this chapter it is shown how these goals can be accomplished for 2D object recognition using a high-dimensional representation space, derived from Fourier domain descriptors of curvilinear features. The associated matching process is between images of these features rather than between invariants derived from the features. For each class of variation, it is shown that a corresponding matching algorithm exists in the representation space and they are all independent of the size of the database.

Recognizing 2D objects in signal space is the concentration of this chapter. In the following chapters the primary problem will shift back to the geometric language that has been developed in previous chapters, and the problem domain will be in Euclidean 3D space.

Chapter 6

Surface Recovery from Curvilinear Features

Information regarding the 3D world can be inferred from sequences of 2D images resulting from the relative motion between the scene and the observer. When the observer does not initiate voluntary movement and is considered stationary relative to an external reference frame, the information from the image sequence is the *optical flow*, which will be studied in the next chapter. The reverse situation when the observer actively navigates through a static environment is the topic of this chapter and Chapter 8.

In this chapter, the focus is on the quantitative constraints imposed by stationary contours on surface shape. It is shown that an active observer can exploit these constraints and move deterministically to the *osculating plane* of a given point on the contour, and from there can recover the normal curvature of the surface as defined by the contour and the associated Frenet frame of the contour. Furthermore, when two non-collinear stationary contours intersect, and one of the principal directions is known, local surface shape can be completely recovered. This is both qualitatively and quantitatively different from the existing methods of recovering surface shape from occluding contours in which a surface parameterization is obtained when the occluding contour slides across the surface [25, 39, 102].

In the first part of this chapter it is proved that the observer can explicitly choose its motion so that the projected curvature of a stationary surface contour monotonically decreases. The lower bound on this projected curvature is reached when the observer reaches the osculating plane defined by the surface contour. During the motion, the observer can either choose to fix the optical axis of the image plane or to rotate the optical axis so that the observed point on the contour is always on the optical axis. The latter is achieved through a combination of camera translation and rotation and is more natural. However, when external references are available, choosing a fixed optical axis properly produces the shortest path to the destination. Both schemes are presented and results proved.

The second part of the chapter describes how the Frenet frame can be recovered once the observer reaches the plane where the projected curvature for the given point on the contour reaches its minimum. In the process of reaching this position the observer can verify if the contour is indeed stationary. Our method for discriminating occluding from stationary contours differs from previous ones in that: (1) no motion parameters are used, and (2) it is applicable within areas where no occluding contours slide across the surface (i.e., elliptic concave parts of the surface). The recovered Frenet frame can be used by the observer to trace the stationary contour in order to recover the same information for all points on the contour (by always staying in the osculating planes). Furthermore, if there are points where two stationary contours intersect, the recovered Frenet frame can be used to parameterize the surface and this parameterization is unique if one of the principal directions (the direction where the normal curvature is either maximum or minimum at the given point) is known. This uniqueness is also true if more than two contours pass close to each other.

Finally, results of various recovered surfaces for a synthetic scene are shown as well as the paths an observer actually takes to reach the osculating plane under purely translational motion, and under translational combined with rotational motion. The validity of the theory is further enforced by showing the results of recovering the surface of a textured vase.

6.1 Theoretical Framework

When a 3D surface is projected into a 2D image plane, the image formed is dependent on the lighting, surface properties, and the location of the image plane. The problem of 3D shape recovery is to describe the surface in a parametric form from a set of 2D images.

In order to parameterize a surface locally, we need to designate two independent basis vectors and an origin. We also need three parameters to characterize how distance (metrics) is measured on the surface and an additional three parameters to measure how the surface tangent turns away from the surface locally. Since these six parameters are not independent, but tied by a set of three *compatibility equations* [31], we need a minimum of three equations relating these parameters to completely characterize the local surface. The problem of recovering surface shape from contours is to derive these equations from a finite number of observations. In this section we describe the surface geometry and imaging model that will be used in the subsequent presentation. We will use $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$ to denote unit basis vectors in the 3D Euclidean coordinate frame (x, y, z).

6.1.1 Curves and Surfaces

Given a reference coordinate frame and a point P on a smooth surface S in space, the local shape of S at P is a set of parameterizations of the form $(x(u, v), y(u, v), z(u, v))$. Each parameterization differs from the others by a rigid transformation (compositions of translation and linear orthogonal transformations). The fundamental theorem of the local theory of surfaces asserts that the first and second fundamental forms uniquely define the surface up to a rigid transformation. These two forms, in turn, can be determined from the surface normal and first and second derivatives of the surface along two principal directions, where the surface curves most and least. This observation is the operational principle for our methods.

For a stationary curve on an object surface, the normal of the curve is always uniquely defined if the curvature κ is not zero. If we determine an orientation for the curve and its tangent direction, the binormal of the curve is then determined by $\hat{\mathbf{b}} = \hat{\mathbf{t}} \times \hat{\mathbf{n}}$. In the following we will use the convention of orienting the observer and the tangent $\hat{\mathbf{t}}$ so that $\hat{\mathbf{x}} \cdot \hat{\mathbf{t}} > 0$ (see Figure 6.1). Hence the direction of $\hat{\mathbf{x}}$ is an "upward" reference direction.

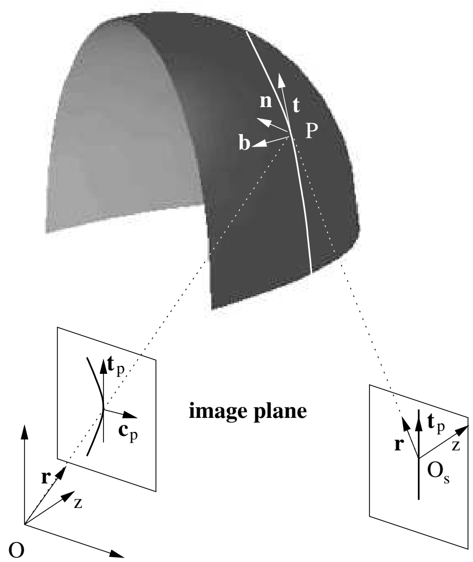


Figure 6.1: Locating the osculating plane for a stationary curve on a convex surface. The plane is reached at location O_s.

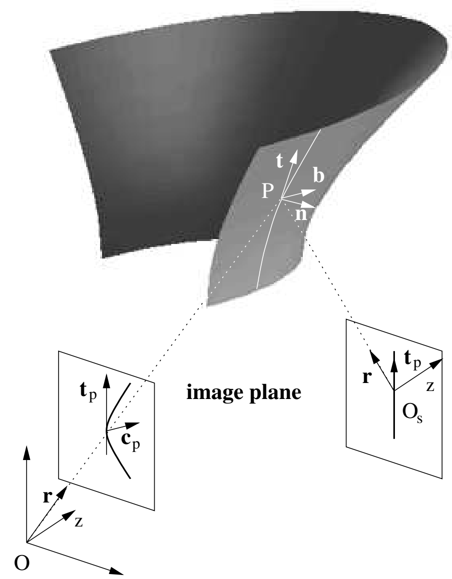


Figure 6.2: Locating the osculating plane for a stationary curve on a concave surface.

6.1.2 Contour Curvature under Projection

Consider the projection of a stationary curve segment C onto the image plane (Figures 6.1 and 6.2). A point P on the curve is represented by the vector $\mathbf{r}$ in the observer frame. Let the Frenet frame at P be $\{\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{b}}\}$, and assume the observer is located at O and looks in the direction (viewing direction) $\mathbf{z}$. The image plane is at $(0, 0, 1)$. The projected curvature κ_p of C_p at point P can be expressed in terms of the Frenet frame $\{\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{b}}\}$ as (see Appendix A.1):

$$\kappa_p = \frac{\kappa |\mathbf{r} \cdot \hat{\mathbf{b}}|}{[(|\mathbf{r} \times \hat{\mathbf{t}})|^2 - (\mathbf{r}, \hat{\mathbf{t}}, \hat{\mathbf{z}})^2)/(\mathbf{r} \cdot \hat{\mathbf{z}})^2]^{3/2}} \tag{6.1}$$

where $(\mathbf{r}, \hat{\mathbf{t}}, \hat{\mathbf{z}})$ is a shorthand for $\mathbf{r} \times \hat{\mathbf{t}} \cdot \hat{\mathbf{z}}$. The advantage of Eq. (6.1) is the use of vector expressions in both the fixed Frenet frame and the observer frame, which is under control of the active observer. These variables are invariant to the observer's frame except for $\mathbf{r}$ and $\hat{\mathbf{z}}$.

Since we haven't distinguished a stationary contour from an occluding contour at this point, the curve could be of either type. In the case of occluding contours, the expression κ_p actually reduces to the equation relating the *geodesic* curvature and the surface normal curvature [25, 59].

From Eq. (6.1) it can be seen that κ_p has minimum value 0 when $\mathbf{r} \cdot \hat{\mathbf{b}} = 0$. This is the case when the observer is in the plane defined by the binormal $\hat{\mathbf{b}}$, called the osculating plane, where the stationary contour C projects onto the image plane as a locally straight line. This is a well known result and we state it formally as follows:

Proposition 6.1.1. Given a spatial curve on a smooth 3D surface and a point on the curve, the minimum projected curvature at this point is zero and will be obtained when the observer is in the osculating plane containing this point.

In the next section we show how an observer can reach this plane (that is, determine the binormal $\hat{\mathbf{b}}$ at a given point on curve C) from any location in space.

6.2 Moving to the Osculating Plane

The formulation above is in the frame of the observer, which we called the *observer frame*. By using the vector representation, we have some advantages when changing the observer frame during motion. However, it can be inconvenient when coordinate transformations are necessary. This difficulty can be circumvented by working in the *object frame* with the origin at P. These two frames can either be related by a pure translation or by a translation plus rotation. In the case of pure translation, the two frames can be transformed back and forth through a translation at any given location of the observer. This case will be referred to as the *translation scheme*. More generally, the observer can choose to orient the observer frame with respect to the object frame in any convenient way and the two frames are related through an arbitrary rigid transformation at a given observer location. This case will be called the *rigid transformation scheme*. Both cases will be analyzed. In the following we will use $\mathbf{r}^*$ to denote the observer's location in the object frame.

6.2.1 Translation Scheme

When an active observer can control its motion so that only pure translation is performed (by using an external reference frame, for example), the observer and object frames are related by $\hat{\mathbf{x}}^* = \hat{\mathbf{x}}$, $\hat{\mathbf{y}}^* = \hat{\mathbf{y}}$, $\hat{\mathbf{z}}^* = \hat{\mathbf{z}}$. Consequently, $\mathbf{r}^* = -\mathbf{r}$. The observer may choose its frame arbitrarily as long as it satisfies $\mathbf{r} \cdot \hat{\mathbf{z}} > 0$ for the selected point on the surface, i.e., the viewing direction $\mathbf{r}$ generally "agrees" with the optical axis $\hat{\mathbf{z}}$. The following definition formalizes the statement that vectors $\mathbf{r}$ and $\hat{\mathbf{z}}$ are pointing generally *in the same direction*.

Definition 6.2.1. *Given a spatial curve C and the Frenet frame $\{\hat{\mathbf{t}}, \hat{\mathbf{n}}, \hat{\mathbf{b}}\}$ at a point on the curve, an observer frame is an* agreeable frame *for the point if for all possible observer movements,* $\mathbf{r}$ *and* $\hat{\mathbf{z}}$ *are in the same octant defined by the Frenet frame.*

Consider κ_p in the object frame as a scalar field of $\mathbf{r}^*$ (i.e., at any given point $\mathbf{r}^*$ in space, there is an associated field value $\kappa_p(\mathbf{r}^*)$) and consider the observer as a detector of the scalar field; that is,

$$\kappa_p(\mathbf{r}^*) = \frac{\kappa |\mathbf{r}^* \cdot \hat{\mathbf{b}}|}{[(|\mathbf{r}^* \times \hat{\mathbf{t}})|^2 - (\mathbf{r}^*, \hat{\mathbf{t}}, \hat{\mathbf{z}})^2)/(\mathbf{r}^* \cdot \hat{\mathbf{z}})^2]^{3/2}}. \tag{6.2}$$

Note that κ_p takes the same form in both the observer and the object frames. Conceptually, κ_p is a quantity to be observed in the observer frame, but in the object frame it is a scalar field defined in three-dimensional space.

Let $\mathbf{c} \stackrel{\triangle}{=} \mathbf{r} \times \hat{\mathbf{t}} = (c_1, c_2, c_3)$. Let's go back to the observer frame and consider how various vectors project to the image plane and analyze their properties. The plane defined by $\mathbf{c}$ intersects the contour on the surface at point P and intersects the image plane along the direction of $\hat{\mathbf{t}}_p$, which is the projection of $\hat{\mathbf{t}}$ on the image plane (see Figure 6.1), and is given by

$$\hat{\mathbf{t}}_p = \frac{(c_2, -c_1, 0)}{(c_1^2 + c_2^2)^{1/2}}. \tag{6.3}$$

The orthogonal direction of $\hat{\mathbf{t}}_p$ is $\mathbf{c}_p$, which is also the projection of the normal of the plane (i.e., $\mathbf{c}$) on the image plane, and is given by

$$\mathbf{c}_p = (c_1, c_2, 0).$$

Let $\mathbf{r}_\perp$ be defined by $\mathbf{r}_\perp \triangleq (-c_2, c_1, (c_2 x - c_1 y)/z)$. Then $\mathbf{r}_\perp$ is orthogonal to $\mathbf{r}$, since $\mathbf{r} \cdot \mathbf{r}_\perp = 0$ and $\mathbf{c}_p$ is orthogonal to $\mathbf{r}_\perp$. Hence $\mathbf{r}_\perp$ is the normal of the plane spanned by $\mathbf{r}$ and $\mathbf{c}_p$.

In Appendix A.2, it is shown that the change of projected curvature κ_p in the direction of $\mathbf{c}_p$ takes the form

$$\nabla \kappa_p \cdot \mathbf{c}_p = \kappa_p \frac{\mathbf{c}_p \cdot \hat{\mathbf{b}}}{\mathbf{r} \cdot \hat{\mathbf{b}}}. \tag{6.4}$$

Note that κ_p is not a differentiable function at $\mathbf{r} \cdot \hat{\mathbf{b}} = 0$, but κ_p^2 is. This is the reason that at $\mathbf{r} \cdot \hat{\mathbf{b}} = 0$, $\nabla \kappa_p \neq 0$.

Since we want to locate the osculating plane, we should move in a direction that reduces κ_p until eventually reaching it. Since κ is bounded below, this is guaranteed if we can always find the desired direction at any given point in space. The osculating plane is defined by $\mathbf{r} \cdot \hat{\mathbf{b}} = 0$ and this plane divides the space outside the object into two regions: $\mathbf{r} \cdot \hat{\mathbf{b}} < 0$ (region I) and $\mathbf{r} \cdot \hat{\mathbf{b}} > 0$ (region II). We will show that, in the translation scheme, the observer can deterministically move either in direction $\mathbf{c}_p$ or $-\mathbf{c}_p$ according to the region the camera is in, in order to reduce κ_p. In particular, we prove the following proposition in Appendix A.3 (see Figures 6.1 and 6.2):

Proposition 6.2.1. For a contour on a convex surface, if the observer chooses an *agreeable frame*, the direction of motion that reduces κ_p in the region defined by $\mathbf{r} \cdot \hat{\mathbf{b}} < 0$ is $\mathbf{c}_p$, and the direction of motion in the region where $\mathbf{r} \cdot \hat{\mathbf{b}} > 0$ is $-\mathbf{c}_p$. For a concave surface, the direction is reversed for each of the regions.

6.2.2 Rigid Transformation Scheme

During active motion, the observer often needs to move by rotating as well as translating. One reason for this type of motion is to adjust the viewing direction so that the surface point being observed is in the direction normal to the image plane. That is,

$$\hat{\mathbf{z}} = \frac{\mathbf{r}}{|\mathbf{r}|}.$$

This case is considered in this section.

Under the above condition, Eq. (6.1) takes the form

$$\kappa_p = \frac{\kappa |\mathbf{r} \cdot \hat{\mathbf{b}}|}{(|\mathbf{r} \times \hat{\mathbf{t}}|^2 / |\mathbf{r}|^2)^{3/2}} = \frac{\kappa |\mathbf{r} \cdot \hat{\mathbf{b}}|}{A^{3/2}}$$

where $A = |\mathbf{r} \times \hat{\mathbf{t}}|^2/|\mathbf{r}|^2$. In the object frame, this becomes

$$\kappa_p(\mathbf{r}^*) = \frac{\kappa |\mathbf{r}^* \cdot \hat{\mathbf{b}}|}{(|\mathbf{r}^* \times \hat{\mathbf{t}}|^2 / |\mathbf{r}^*|^2)^{3/2}} = \frac{\kappa |\mathbf{r}^* \cdot \hat{\mathbf{b}}|}{[|\mathbf{c}^*|^2/|\mathbf{r}^*|^2]^{3/2}}. \tag{6.5}$$

At each new camera position, assume the camera also rotates so that the direction of $\hat{\mathbf{z}}$ is coincident with $\mathbf{r}$. The value of κ_p, then, depends only on $\mathbf{r}$, $\hat{\mathbf{t}}$ and $\hat{\mathbf{b}}$. Since the gradient vector needs to be computed in order to determine the dependency between the motion direction and the change of κ_p, and the object frame is the only one where we can perform the gradient operation, we have to use Eq. (6.5) directly in its general component form, that is,

$$\kappa_p = \frac{\kappa|\mathbf{r}^* \cdot \hat{\mathbf{b}}|}{[(c_1^{*2} + c_2^{*2} + c_3^{*2})/(x^{*2} + y^{*2} + z^{*2})]^{3/2}}. \tag{6.6}$$

It then can be shown that

$$\nabla\kappa_p = \pm\frac{\kappa}{A^{5/2}}\left[A\hat{\mathbf{b}} - 3\frac{(\mathbf{r}^* \cdot \hat{\mathbf{b}})}{|\mathbf{r}^*|^2}(\mathbf{c}^* \times \hat{\mathbf{t}}) - \frac{|\mathbf{c}^*|^2}{|\mathbf{r}^*|^2}\mathbf{r}^*\right]. \tag{6.7}$$

Since $\mathbf{c}^* \cdot (\mathbf{c}^* \times \hat{\mathbf{t}}) = 0$ and $\mathbf{c}^* \cdot \mathbf{r}^* = 0$ we have

$$\nabla\kappa_p \cdot \mathbf{c}^* = \pm\frac{\kappa}{A^{3/2}}(\mathbf{c}^* \cdot \mathbf{b}).$$

Using Eq. (6.5), we have

$$\nabla\kappa_p \cdot \mathbf{c}^* = -\nabla\kappa_p \cdot \mathbf{c} = \kappa_p\frac{\mathbf{c}^* \cdot \hat{\mathbf{b}}}{\mathbf{r}^* \cdot \hat{\mathbf{b}}}. \tag{6.8}$$

We can then prove a proposition similar to the one for the translation scheme:

Proposition 6.2.2. For a contour on a convex surface, if the observer chooses its frame so that $\hat{\mathbf{z}} = \mathbf{r}/|\mathbf{r}|$, the direction of motion that reduces κ_p in the region defined by $\mathbf{r} \cdot \hat{\mathbf{b}} < 0$ is $\mathbf{c}$, and the direction of motion in the region where $\mathbf{r} \cdot \hat{\mathbf{b}} > 0$ is $-\mathbf{c}$. For a concave surface, the direction is reversed in each region.

Note that in the observer frame, $\mathbf{c}$ is actually orthogonal to $\hat{\mathbf{z}}$ since $\hat{\mathbf{z}}$ and $\mathbf{r}$ are coincident. Hence $\mathbf{c} = \mathbf{c}_p$ and the above two propositions become identical. The only difference is that for the translation scheme $\mathbf{c}_p$ is always on the same plane, while in the rigid transformation scheme $\mathbf{c}$ translates and rotates as the observer approaches the osculating plane.

6.2.3 Discussion

We have shown that the observer can always move deterministically in either region outside the object surface to reach the osculating plane for a given marked point on the object surface. The direction of movement, $\mathbf{c}_p$, is always orthogonal to the projected tangent, $\mathbf{t}_p$. In the case of pure translation, the observer always moves within the established image plane and the point where it reaches the osculating plane will be on the intersecting line of the image plane and the osculating plane. Hence the length of path from the initial location to the osculating plane is determined by the initial viewing direction. Furthermore, without an external landmark, it is very difficult to verify if the motion is purely translational. This is not the case for the rigid transformation scheme since the surface mark itself is the external reference. The observer also has better control over the path from the initial position to the osculating plane because the viewing direction can be guided by the reference rather than arbitrarily chosen. This difference is shown in Section 6.5.

If the marked point becomes occluded during the observer motion, an alternative path has to be found. This is generally achieved by choosing a different direction of motion while keeping the projected curvature κ_p constant. During this action, the observer essentially "moves around" the obstacle while keeping same "distance" from the marked point (see Chapter 8). This will be the real Euclidean distance when the marked point is on a constant curvature surface.

On the other hand, when there are no well-defined stationary marks on the contour, point-wise correspondence across observations becomes problematic and the rigid transformation scheme cannot be used. However, if the observer motion can be assured to be translational only (by external reference, for example), the plane formed by the initial observation direction, $\mathbf{r}$, and the direction of movement, $\mathbf{c}_p$, will intersect the object surface at the point P. In this case, the translation scheme will be the only one applicable.

Once in the osculating plane the surface binormal $\hat{\mathbf{b}}$ can be recovered by

$$\hat{\mathbf{b}} = \pm\frac{\mathbf{r}\times\hat{\mathbf{t}}_p}{|\mathbf{r}\times\hat{\mathbf{t}}_p|}. \tag{6.9}$$

The sign of the above expression is determined by the local shape of the surface along the contour (negative for convex and positive for concave).

Propositions 6.2.1 and 6.2.2 also provide a way to determine the shape of the surface along the contour qualitatively, i.e., if it is a convex or concave surface strip [26, 60]. For example, if the projection of the contour is convex to the right (open to the left) and moving right decreases κ_p, then the surface strip is convex; otherwise it is concave.

Based on an expression similar to Eq. (6.1), Cipolla [26] derived qualitative results regarding surface shape. He also showed that if image velocity can be computed accurately, the deformation of stationary contours can be used to compute the curvature of the contours and to constrain the viewer motion.

Next we show how the rest of the Frenet frame can be recovered.

6.3 Frenet Frame Recovery

Once we have found the osculating plane, the recovery of the rest of the Frenet frame becomes possible. Consider Figure 6.3.

6.3.1 Curvature

If the observer chooses its frame so that $\hat{\mathbf{z}} = -\hat{\mathbf{b}}$, then as long as the observer translates only along $\hat{\mathbf{b}}$, we always have $\hat{\mathbf{t}} = (x', y', 0)$ and $\mathbf{c} = \mathbf{r}\times\hat{\mathbf{t}} = (-yz', zx', 0)$. Hence Eq. (A.4)

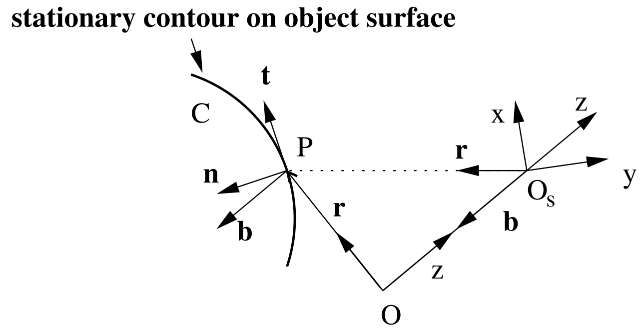


Figure 6.3: Recovery of Frenet vectors when moving away from the osculating plane along the binormal direction.

becomes

$$\kappa_p = \frac{\kappa|\mathbf{r}\cdot\hat{\mathbf{z}}|}{(x'^2 + y'^2)^{3/2}} = \frac{\kappa|\mathbf{r}\cdot\hat{\mathbf{z}}|}{|\hat{\mathbf{t}}|^3} = \kappa|\mathbf{r}\cdot\hat{\mathbf{z}}|. \tag{6.10}$$

Consequently, if the observer moves in the direction along $\hat{\mathbf{b}}$ a distance d, the contour curvature κ at P is

$$\kappa = \frac{\kappa_p}{d}.$$

In this process, the exact measurement system used by the observer to measure the distance d is not important as long as the projected curvature κ_p is measured against the same system. However, if the contour curvature κ is used to recover surface shape (see below), the measurement system has to be consistent with the metric used for the surface.

It should be noted that the requirement that the observer translate strictly along $\hat{\mathbf{b}}$ implicitly assumes the existence of some external references. In another words, this action cannot be "intrinsic" and some kind of external reference must be used to accomplish this motion.

6.3.2 Tangent and Normal Vectors

From Eq. (6.3) we get

$$\hat{\mathbf{t}}_p = \frac{(c_2, -c_1, 0)}{(c_1^2 + c_2^2)^{1/2}} = (x', y', 0). \tag{6.11}$$

Hence the components of $\hat{\mathbf{t}}$ are identical to the components of $\hat{\mathbf{t}}_p$ in the selected observer frame. This result gives us the tangent. Finally, from $\hat{\mathbf{n}} = \hat{\mathbf{b}} \times \hat{\mathbf{t}}$ we get the last member of the Frenet frame, the normal.

Hence the Frenet frame can be recovered by observing the deformation of geometric invariants (the curvature of the projected contour) without knowing the depth z, which can be recovered from the triangulation of O_sOP in Figure 6.3 if external reference points can be found. Next we show that all the geometric metrics for the space curve C can be recovered.

6.3.3 Torsion

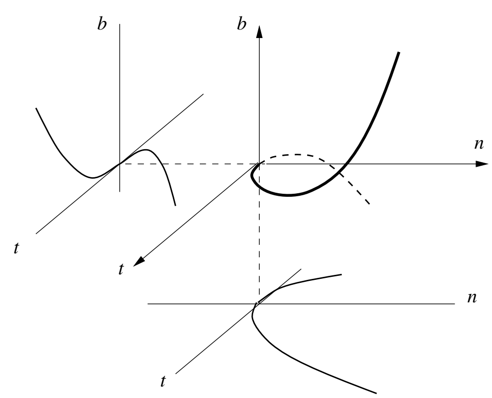


Figure 6.4: Curve projection onto planes defined by the Frenet frame.

From the Serret-Frenet equations (Eq. (3.10)), if the coordinate system is chosen to be the

Frenet frame (i.e., $\hat{\mathbf{x}} = \hat{\mathbf{t}}$, $\hat{\mathbf{y}} = \hat{\mathbf{n}}$, $\hat{\mathbf{z}} = \hat{\mathbf{b}}$), we can derive the *local canonical form* of C :

$$\begin{aligned} x(s) &= s - \frac{\kappa^2 s^3}{6} + R_x \\ y(s) &= \frac{\kappa s^2}{2} + \frac{\kappa' s^3}{6} + R_y \\ z(s) &= -\frac{\kappa \tau s^3}{6} + R_z. \end{aligned} \tag{6.12}$$

The projections of C onto the $\hat{\mathbf{t}}-\hat{\mathbf{n}}$ and $\hat{\mathbf{t}}-\hat{\mathbf{b}}$ planes are shown in Figure 6.4. As can be seen in the equation, the local form of C on the $\hat{\mathbf{t}}-\hat{\mathbf{b}}$ plane is cubic and by estimating the third-order slope across the origin we can compute τ. This completely determines curve C at the point.

It should be noted that we cannot actually use the chosen $\hat{\mathbf{z}}$ as our observer frame since the requirement $\hat{\mathbf{z}} = \hat{\mathbf{b}}$ will generally put the observer at a side view of the surface where the stationary contour coincides with the occluding contour. On the other hand, if we choose $\hat{\mathbf{z}} = \hat{\mathbf{n}}$ then the slope across the $\hat{\mathbf{x}} - \hat{\mathbf{y}}$ (i.e., $\hat{\mathbf{b}} - \hat{\mathbf{t}}$) plane is zero (i.e., $\kappa_p = 0$), making the estimation of τ unreliable. However, if the goal is to recover the surface geometry of the object, then recovering the curvature and the Frenet frame (actually $\hat{\mathbf{t}}$) is sufficient.

6.4 Applications

6.4.1 Distinguishing Stationary Contours from Occluding Contours

There are qualitative differences between the deformations of a stationary and an occluding contour when the observer can move in controlled ways [69]. These differences enable us to discriminate between these two types of contours without first recovering the surface shape.

The deformation of a stationary contour occurs for two reasons. The first kind of deformation occurs during the process of locating the osculating plane, when the contour, as projected onto the image plane, deforms according to Propositions 6.2.1 and 6.2.2. The second kind occurs when the observer moves along the binormal $\hat{\mathbf{b}}$ after the osculating plane is identified. In

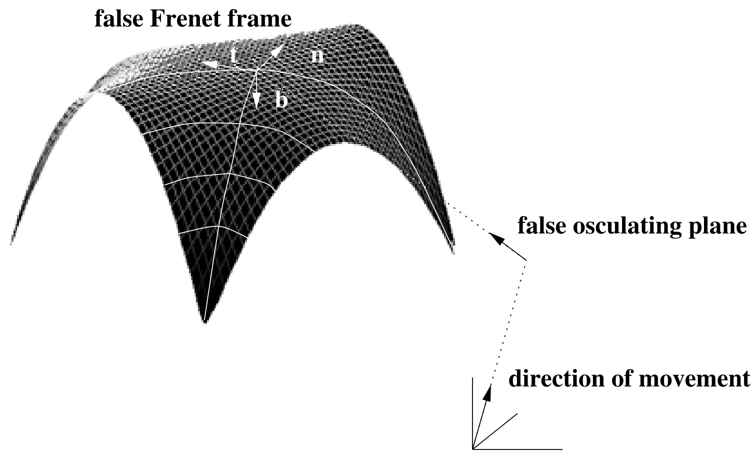


Figure 6.5: An occluding contour that appears to be a stationary contour.

the second form, the contour does not actually deform locally (see Eq. (6.11)). There could be a surface and a motion that makes the deformation of the occluding contour behave like a stationary contour in the first form (see Figure 6.5). However, *all* occluding contours must deform when the observer moves along the (false) binormal because this direction is actually toward the object surface (see Eq. (6.9) and Figure 6.5).

In the previous sections we handled the case where there are no other markings on the surface. That is, there is only one contour with a marked point in the region of interest. In practice, additional contours, texture or surface markings will make the task much easier. Nonetheless, we show that in this worst case, the deformation of projected curvature, κ_p, and the Frenet frame carry with them *intrinsic* information that allows us to distinguish stationary contours from non-stationary ones.

6.4.2 Surface Shape Recovery from Multiple Contours

For a given local parameterization, the six parameters in the first and second fundamental forms that satisfy the three *compatibility equations* [31] completely determine the surface shape up to a rigid transformation. This is the fundamental theorem of the local theory of surfaces. Hence, there are three degrees of freedom that need to be fixed in order to determine the local surface shape. Since the value of the second fundamental form along a given direction equals the normal curvature of the surface along that direction, two intersecting stationary contours provide a local parameterization and two constraints for the three degrees of freedom. In this section, we consider how the third constraint can be obtained.

6.4.2.1 Surface Shape from Principal Curvature

From the differential theory of surface geometry, the two principal directions where normal curvature for the surface reaches extrema are orthogonal to each other. Any normal curvature κ_n at the point relates to these two extremal curvatures by *Euler's formula:*

$$\kappa_n = \kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta$$

where κ_1 and κ_2 are the two principal curvatures and θ is the angle between the tangent for κ_n and one of the principal directions. If one of the stationary contours is along the principal direction, we can solve for both principal directions and the associated extreme curvatures. When neither of the two contours is along a principal direction but one of the principal directions can be determined by other means, the two Euler equations relating $\kappa_1, \kappa_2, \theta$ and the two known normal curvatures along the two contours enable us to solve the surface geometry completely.

Since principal directions are directions where the normal section of the surface is maximally or minimally curved, we need an active procedure capable of inspecting all directions around the surface point. *Shape from occluding contour* methods work well when the surface is elliptically convex in the neighborhood of the point, but fail otherwise.

6.4.2.2 Surface Shape from another Contour

If we cannot determine any of the principal directions by examining the surface visually, they can still be determined algebraically. The result can be exact if there is another stationary contour passing through the intersection point where the two contours intersect. In most cases, the shape can be estimated when additional contours pass by in the vicinity of the point. In the first case, we have three Euler equations to solve all required parameters. In the second case, since the third contour will intersect at least one of the first two contours, we can *parallel transport* [31, 60] the tangent and curvature along this third contour from the additional intersection point to the first intersection point and then solve the set of three Euler equations. The accuracy of this parallel transport depends on the curvature of the surface along which we make the parallel transport. For a locally cylindrical surface the result will be exact.

6.4.2.3 Mesh Representation

Stationary contours as visual cues are most effective in "meshed" representations of surfaces. For example, a meshed representation of a synthetic surface is shown in Figure 6.7. Intersecting lines with zero projected curvature along a particular direction give cues of a locally flat surface in that direction. Consequently, by assuming the surface is locally parabolic or cylindrical in one direction, the mesh lines in the orthogonal direction provide the needed deformation as cues of surface shape in that direction. This is achieved through the implicit assumption that the orthogonal direction has constant curvature and the mesh lines in that direction provide a deformed sequence exactly as if the observer moves along that direction and observes the deformation of a stationary contour.

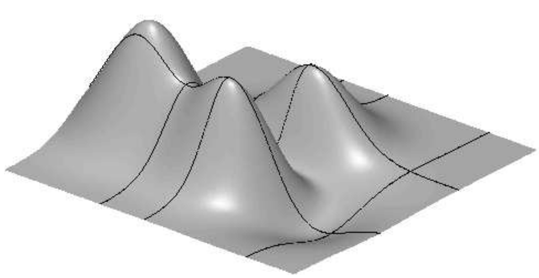


Figure 6.6: A synthetic surface with stationary contours.

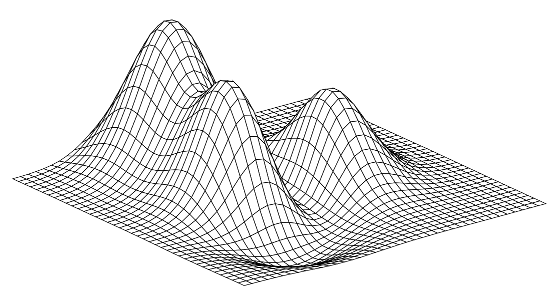


Figure 6.7: Mesh representation of the surface.

6.5 Examples

In this section both a synthetic surface and a ray-traced vase are used to show results of how the theory can be implemented. The surface in Figure 6.6 contains peaks, valleys and saddles and the dark contours on the surface are stationary contours and will be used by the observer to recover surface shape at the intersection points.

The recovery process involves camera motions relative to both intersecting contours and, for each contour, recovering the tangent $\hat{\mathbf{t}}$ and the curvature κ. From these two tangents, $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$, the surface normal can be found as

$$\hat{\mathbf{N}} = \frac{\hat{\mathbf{t}}_1 \times \hat{\mathbf{t}}_2}{|\hat{\mathbf{t}}_1 \times \hat{\mathbf{t}}_2|}.$$

The normal curvature κ_n of the surface along $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$ can be recovered by applying the formula

$$\kappa_n = \kappa \hat{\mathbf{n}} \cdot \hat{\mathbf{N}}$$

where $\hat{\mathbf{n}}$ is the normal vector in the Frenet frame for the contours.

Since we did not attempt to find the directions of the principal curvatures, it is assumed that these two contours are actually in the principal directions. Under this assumption, all six parameters of the first and second fundamental forms can be computed [31] and we can parameterize the local surface by using the two tangents as two basis vectors and the intersection point as the origin. To illustrate, we overlay on the surface members of the family of quadratic functions that have the same parameters at the given intersecting points in Figure 6.8. This is accomplished by using a quadratic Monge patch parameterization $(u, v, h(u, v))$ and solving for all the coefficients using the parameters from the two fundamental forms. The resulting functions are translated to the surface point and rotated according to the recovered surface normal $\hat{\mathbf{N}}$. Locally, the quadratic functions exactly match the surface. Globally, the degree of fit will be determined by the variation of the normal curvatures along the principal curves passing by the surface point.

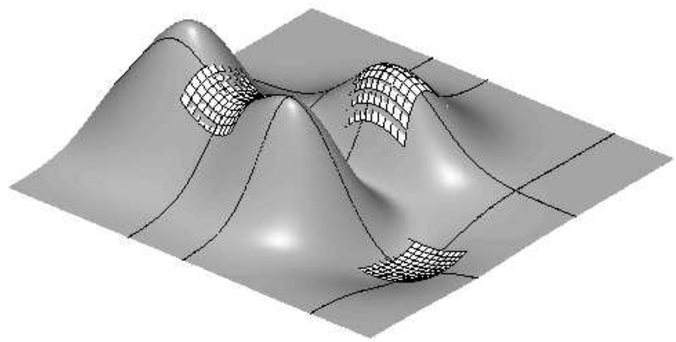


Figure 6.8: Synthetic surface with recovered elliptic and hyperbolic surface patches.

Figure 6.9 shows the paths taken by the camera in the translation and the rigid transformation schemes starting from the same initial location. The scene is set up so that $\hat{\mathbf{z}}$ is in the upward direction and $\{\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}\}$ forms a right-handed system. At the surface point being tracked, the stationary contour moves in the direction of $\hat{\mathbf{t}}$. The image plane for the translation scheme is arbitrarily set to be in the $-\hat{\mathbf{z}}$ direction.

It can be seen that the path taken by the translation scheme is parallel to the x-y plane and intersects the osculating plane horizontally, i.e., the observer moves completely within the image plane. On the other hand, the rigid transformation scheme takes a more curved path because of the gradual turning of the viewing direction. In this particular case, the translation scheme actually reaches the osculating plane faster because of the position of the image plane. If the observer can ensure that its movement is purely translational by referring to external references, the path will be the shortest of the two schemes in all cases. However, this shortest path will bring the observer directly to the surface, i.e., the path intersects the osculating plane at the surface. This may not be desirable in practical cases.

A more realistic example is shown in Figure 6.10 in which various stationary contours on a vase are tracked and the local surface shape at intersecting stationary marks is recovered. Complete shape recovery is possible only at these stationary marks. For points along a contour

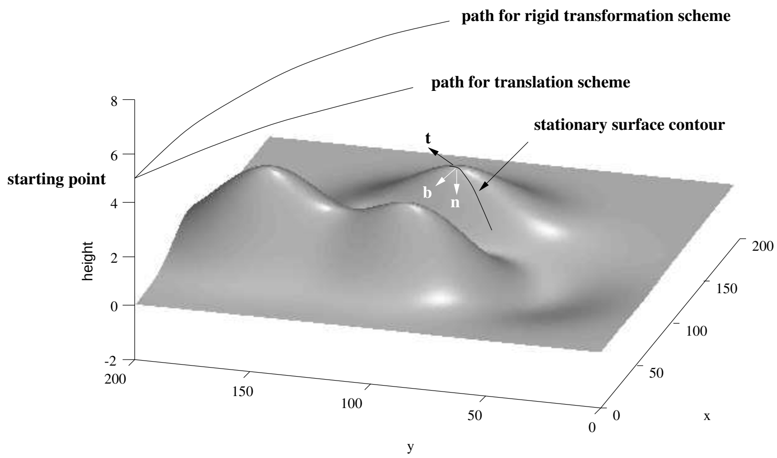


Figure 6.9: Paths produced by the translation scheme and the rigid transformation scheme.

---

between two marks, interpolation is used to estimate the curvature in the direction orthogonal to the contour.

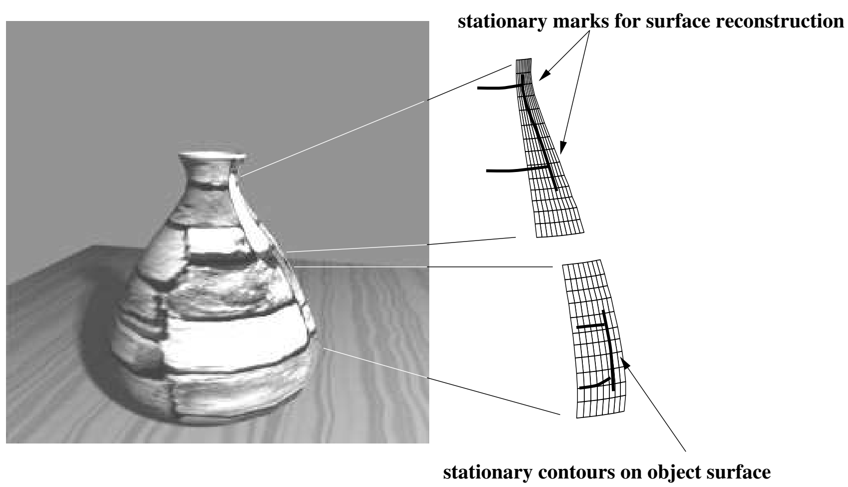


Figure 6.10: Surface recovery from stationary contours and marks.

6.6 Summary

Curvilinear features on object surfaces are useful for constraining surface shape. In contrast to some other kinds of contours, stationary contours do not provide two-dimensional constraints on surface shape under observer motion. However, from the constraints they do provide, we have shown that the surface in the neighborhood of the contour can be recovered without knowing or being sensitive to measurement errors in both observer and image motion. All the parameters characterizing a stationary contour as a spatial curve can be recovered by controlled motion of the observer. In the process, the problem of discriminating between occluding contours and stationary contours is also solved. Another major result is the analysis leading to a method an observer can use to reach the osculating plane. Depending on whether external references other than the surface point are available or not, two different strategies for observer motion were presented. They perform differently but both enable the observer to move in directions that monotonically decrease the projected contour curvature on the image plane.

Contrary to the belief that stationary contours are only useful for acquiring qualitative surface information, we demonstrated that curvilinear features on a surface can be considered as "samples" of surface shape and as long as there are enough features in a region of interest, the shape can be recovered quite accurately. Suggestions are given concerning how to acquire information about the principal directions so that two intersecting surface contours can be used to constrain the local surface completely. Error analysis for the case when we can parallel transport a nearby point to the intersection point in order to recover the surface geometry is an important subject for future study.

The active navigation necessary for the observer to acquire important information in order to solve the problem of surface geometry will be further developed in Chapter 8, in which the information content is much expanded and is not restricted to static curvilinear features as are treated in this chapter.

Chapter 7

Computation and Segmentation of Optical Flow

Motion perception is essential in both shape recovery and navigation, but, for an observer, the computation of a 2D motion field can only be carried out through observing the optical flow which is the projection of the motion field onto the image plane. This correspondence between motion field and optical flow is not one-to-one, even though they are coincident for most cases [51]. In the case of local shape recovery, the inverse problem can be solved if the surface is smooth and high accuracy measurement can be achieved for second-order variations of the optical flow. The measurements of optical flow and its spatial variation are generally noisy and hard to make accurate. On the other hand, the smoothness of the optical flow has a direct correspondence in the motion field and can be computed by using only the first order variation of the optical flow. The determination of the discontinuous boundary of an optical flow field is the segmentation of optical flow. Discontinuities in optical flow correspond directly to discontinuities of either the surface shape or surface orientation.

In this chapter, the focus is on the geometric properties of optical flow and the segmentation of it. A new method for computing optical flow from a spatio-temporal image volume is presented. It is shown that kernels that are local in both the spatial and temporal domains can be designed to compute the optical flow.

Once the optical flow field is computed, we show that an observer can utilize its mobility to actively control the shape of the optical flow field, which directly reflects the surface shape of the object. Furthermore, we present an algorithm for segmenting an optical flow field, which can be proved to be correct under smooth observer motion.

This chapter starts with the decomposition of a general vector field into divergence, curl and deformation fields. Each provides information on the object surface relative to the observer. This information can be used by the active observer to control its motion in order to complete relevant tasks (e.g., navigation or shape recovery). Following this, a spatio-temporal receptive field is introduced in order to compute the optical flow using a polar parameterization. It is shown next how the eigenvalues of the optical flow decomposition can be related to the observer-controlled motion and surface geometry. Finally, a new method of optical flow decomposition is presented for an observer-controlled translation. This is achieved by measuring the directional derivative of the magnitude of optical flow in the direction of the flow itself.

7.1 Theoretical Framework: 2D Vector Field Decomposition

On a 2D Euclidean manifold (ξ, η) the *integral curves* of a 2D linear vector field $\mathbf{u} = (\mu, \nu)$ are the family of curves $\mathbf{q} = (\xi(s), \eta(s))$ defined by

$$\begin{aligned} \mu &= \frac{\partial \xi}{\partial s} = \frac{\partial \mu}{\partial \xi}\xi + \frac{\partial \mu}{\partial \eta}\eta \\ \nu &= \frac{\partial \eta}{\partial s} = \frac{\partial \nu}{\partial \xi}\xi + \frac{\partial \nu}{\partial \eta}\eta \end{aligned} \tag{7.1}$$

where s is curve length. Let the matrix $\mathbf{P}$ be

$$\mathbf{P} = \begin{pmatrix} \frac{\partial \mu}{\partial \xi} & \frac{\partial \mu}{\partial \eta} \\ \frac{\partial \nu}{\partial \xi} & \frac{\partial \nu}{\partial \eta} \end{pmatrix}.$$

Eq. (7.1) can then be written in the form $\mathbf{u}^T = \mathbf{P}\mathbf{q}^T$. For a general vector field $\mathbf{u}$, Eq. (7.1) provides the first-order approximation to $d\mathbf{u}$ (by Taylor series) in the form $d\mathbf{u}^T = \mathbf{P}\, d\mathbf{q}^T$.

$\mathbf{P} = (p_{ij})$ can be decomposed into the sum of a symmetric matrix $\mathbf{P}^s = (p^s_{ij})$ and an antisymmetric matrix $\mathbf{P}^a = (p^a_{ij})$ according to $p^s_{ij} = (p_{ij} + p_{ji})/2$ and $p^a_{ij} = (p_{ij} - p_{ji})/2$:

$$\mathbf{P}^s = \frac{1}{2}\begin{pmatrix} 2\frac{\partial \mu}{\partial \xi} & \frac{\partial \mu}{\partial \eta} + \frac{\partial \nu}{\partial \xi} \\ \frac{\partial \mu}{\partial \eta} + \frac{\partial \nu}{\partial \xi} & 2\frac{\partial \nu}{\partial \eta} \end{pmatrix}$$
$$\mathbf{P}^a = \frac{1}{2}\begin{pmatrix} 0 & \frac{\partial \mu}{\partial \eta} - \frac{\partial \nu}{\partial \xi} \\ \frac{\partial \nu}{\partial \xi} - \frac{\partial \mu}{\partial \eta} & 0 \end{pmatrix}. \tag{7.2}$$

Since a symmetric matrix can always be diagonalized by a similar transform, $\mathbf{P}^s$ can be put into the form

$$\mathbf{P}^s = \mathbf{Q}^{-1}\begin{pmatrix} \zeta_1 & 0 \\ 0 & \zeta_2 \end{pmatrix}\mathbf{Q}$$

where $\zeta_1 > \zeta_2$ and $\mathbf{Q}$ is an orthogonal matrix with $|\mathbf{Q}| = 1$. This transform has the property

$$tr\ \mathbf{P}^s = \frac{\partial \mu}{\partial \xi} + \frac{\partial \nu}{\partial \eta} = \zeta_1 + \zeta_2$$

Let

$$\mathbf{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{J}_2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \mathbf{K}_2 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Then, by the above property, we have

$$\begin{aligned}
2\mathbf{P}^s &= 2\mathbf{Q}^{-1}\begin{pmatrix}\zeta_1 & 0\\ 0 & \zeta_2\end{pmatrix}\mathbf{Q}\\
&= \mathbf{Q}^{-1}\left[(\zeta_1+\zeta_2)\mathbf{I}_2 + (\zeta_1-\zeta_2)\mathbf{J}_2\right]\mathbf{Q}\\
&= (\frac{\partial\mu}{\partial\xi} + \frac{\partial\nu}{\partial\eta})\mathbf{I}_2 + (\zeta_1-\zeta_2)\mathbf{Q}^{-1}\mathbf{J}_2\mathbf{Q}.
\end{aligned} \tag{7.3}$$

A more compact form can be reached by noting that if the 2D vector field $\mathbf{u} = (\mu, \nu)$ is treated as a 3D field $(\mu, \nu, 0)$ then

$$\begin{aligned}
\nabla\cdot\mathbf{u} &= \frac{\partial\mu}{\partial\xi} + \frac{\partial\nu}{\partial\eta}\\
\nabla\times\mathbf{u} &= (\frac{\partial\nu}{\partial\xi} - \frac{\partial\mu}{\partial\eta})\hat{\mathbf{e}}_z.
\end{aligned} \tag{7.4}$$

If we denote the only component of the curl as $(\nabla\times\mathbf{u})_z$, the matrix $\mathbf{P}$ has the decomposed form:

$$\mathbf{P} = \frac{1}{2}(\nabla\cdot\mathbf{u})\mathbf{I}_2 + \frac{1}{2}(\nabla\times\mathbf{u})_z\mathbf{K}_2 + \frac{1}{2}(\zeta_1-\zeta_2)\mathbf{Q}^{-1}\mathbf{J}_2\mathbf{Q}. \tag{7.5}$$

Following [62] we will refer these three decomposed components of **u** as the *divergence, curl* and *deformation* fields, respectively.

7.2 Properties of the Decomposed Fields

Given the set of differential equations for integral curves as in Eq. (7.1), we can characterize the curves by the *eigenvalues* λ_k of the matrix $\mathbf{P}$ (see, e.g., [15]) defined by the equation $\mathbf{P}\mathbf{v}_k =$

$\lambda_k \mathbf{v}_k$, where $\mathbf{v}_k$ is the *eigenvector* of $\mathbf{P}$. The basis of the solution of Eq. (7.1) is then

$$\begin{pmatrix} \xi \\ \eta \end{pmatrix} = e^{\lambda_1 s}\mathbf{v}_1,\ e^{\lambda_2 s}\mathbf{v}_2$$

where λ_k is given by solving the *characteristic equation*

$$|\mathbf{P} - \lambda \mathbf{I}_2| = 0. \tag{7.6}$$

Since Eq. (7.5) is the algebraic sum of three vector sub-fields, the properties of the vector field are completely defined by these three sub-fields.

7.2.1 Divergence Field

The characteristic equation for the divergence field

$$\frac{1}{2}(\nabla \cdot \mathbf{u})\mathbf{I}_2$$

is

$$\lambda^2 - 2\lambda + 1 = 0$$

which defines a degenerate integral curve in a “star” configuration (see Figure 7.1), since two eigenvalues are real and identical. For optical flow, the divergence field is a result of the observer moving toward or away from the object, and provides information regarding *time to contact*.

7.2.2 Curl Field

The characteristic equation for the curl field

$$\frac{1}{2}(\nabla \times \mathbf{u})_z \mathbf{K}_2$$

is

$$\lambda^2 + 1 = 0$$

which defines an integral curve in a "vortex" configuration (see Figure 7.1), since two eigenvalues are pure imaginary and opposite in sign. For optical flow, this is a pure rotation with surface normal coincident with the line of sight.

7.2.3 Deformation Field

Since the deformation field is defined by

$$\frac{1}{2}(\zeta_1 - \zeta_2) Q^{-1} \mathbf{J}_2 Q$$

with $|\mathbf{Q}| = 1$ we can treat $\mathbf{Q}$ as a rotation, i.e.,

$$\mathbf{Q} = \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix}$$

and the characteristic equation is

$$\lambda^2 - 1 = 0$$

which defines an integral curve of a "saddle point" configuration (see Figure 7.1). The two eigenvalues are real and opposite in sign. For optical flow, the deformation field carries information on surface orientation. Since when $\zeta_1 = \zeta_2$, the deformation field disappears, the observer can use the shape of the deformation field to control its motion with respect to a reference area on the object surface (see Eq. (7.20)).

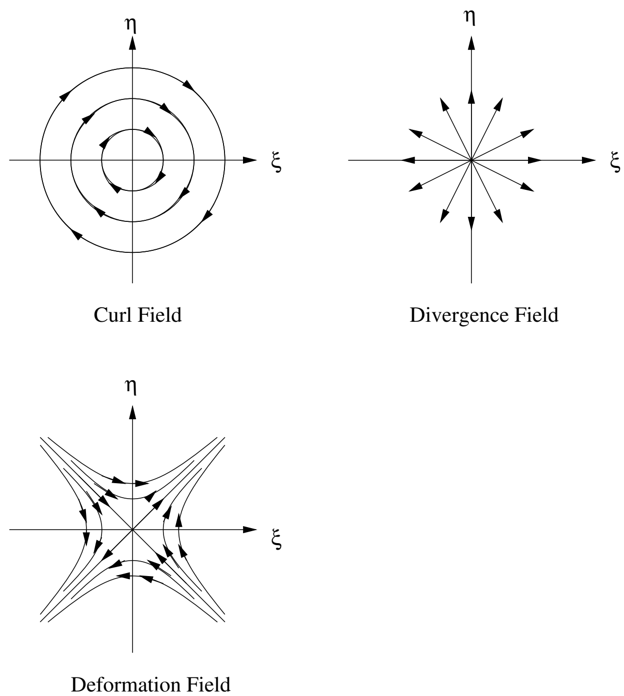


Figure 7.1: Integral curves of Vector fields corresponding to decomposed sub-fields

7.3 Properties of Integrated Fields

The integrated vector field is determined by three canonical subfields formulated in Eq. (7.5). Its properties can be investigated by examining the eigenvalues of $\mathbf{P}$. The eigenvalues are themselves functions of the image coordinates (ξ, η) defined at each point on the image plane, as is the field itself.

If we insert the orthogonal matrix $\mathbf{Q}$ and let $c = (\nabla \times \mathbf{u})_z/2,\ d = (\nabla \cdot \mathbf{u})/2,\ e = (\zeta_1 - \zeta_2)/2,$

P can be written in the form

$$\begin{aligned}\mathbf{P} &= \frac{d}{2}\mathbf{I}_2 + \frac{c}{2}\mathbf{K}_2 + \frac{e}{2}\begin{pmatrix}\cos 2\gamma & -\sin 2\gamma \\ -\sin 2\gamma & -\cos 2\gamma\end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix}d + e\cos 2\gamma & -c - e\sin 2\gamma \\ c - e\sin 2\gamma & d - e\cos 2\gamma\end{pmatrix}.\end{aligned}$$

Hence the characteristic equation for **P** is

$$\lambda^2 - 2d\lambda + (c^2 + d^2 - e^2) = 0$$

and the eigenvalues are $\lambda = d \pm (e^2 - c^2)^{1/2}$. From this we can make the observation that $e^2 - c^2$ (curl and deformation) acts as an essential factor in deciding the field characteristics. One of the interesting cases is when the curl and deformation fields cancel each other so that only the divergence field shows up. This is different from vanishing curl and deformation fields, but it appears identical to the observer. The relationship between eigenvalues and observer motion will be derived in Section 7.5.

7.4 Optical Flow Computation

The receptive field framework developed for static images can readily be extended to the computation of optical flow (see Eqs. (4.2) and (4.3)). The primary difference is that the optical flow field are localized vectors with both magnitude and orientation, while image contours have only localized orientation, as we will see shortly.

Consider the spatio-temporal volume $I(\mathbf{x}, t)$ of the image in the spatially local region N_P of the point P and temporally local region $t \in [0, 1]$ when N_P is moving with image velocity $\mathbf{u} = (\mu, \nu) = (\rho\cos\theta, \rho\sin\theta)$, where ρ is the absolute velocity and θ is the direction of the optical flow. The trace of the point in $I(\mathbf{x}, t)$ will be the line $\mathbf{x} = t\mathbf{u}$, i.e., the locus of all the possible traces will be on the cone (see Figure 7.2) $|\mathbf{x}| = \rho t$.

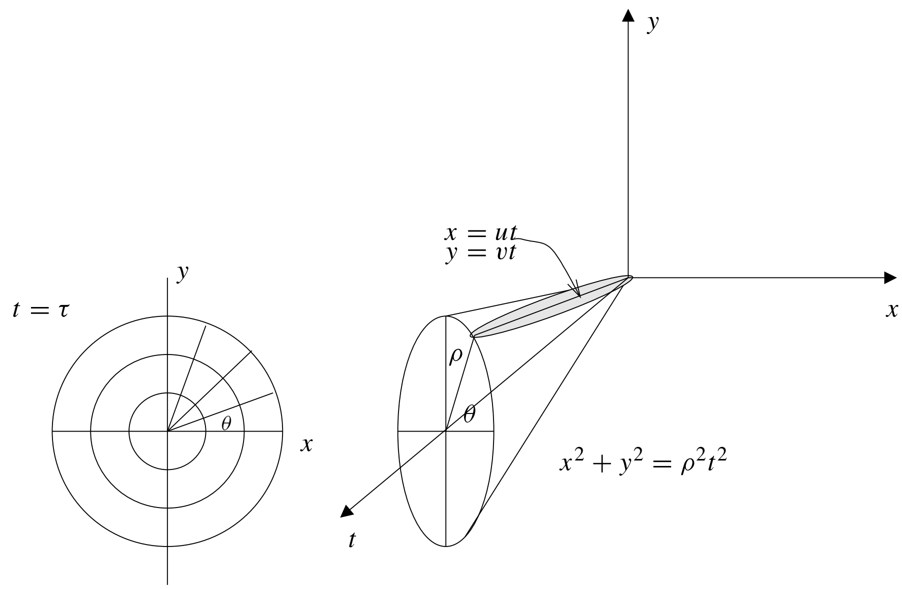


Figure 7.2: Optical flow in $\mathbf{x} - t$ frame

Let $\mathbf{x}' = \mathbf{x} - t\mathbf{u}$. Since the energy of the spatio-temporal line will be confined entirely within the neighborhood of the line, we can consider a spatio-temporal receptive field of the following form for $t \in [0, 1]$:

$$\psi_0^m(\mathbf{x}, \rho, \theta, t; \sigma) = (1 - \frac{|\mathbf{x}'|^2}{\sigma^2}) \exp\left[-\frac{|\mathbf{x}'|^2}{2\sigma^2}\right]. \tag{7.7}$$

This receptive field has maximum response when the feature has a radius of 2σ and the spatio-temporal volume is exactly $\mathbf{x} = t\mathbf{u}$. Additionally, its symmetric shape and vanishing volume indicate that it responds only to feature points of size smaller than the specified size.

Since the local temporal scope is in [0, 1], the temporal energy of the receptive field will always be finite. Hence, the problem of computing optical flow at $\mathbf{x} = (0, 0)$ is to find ρ and θ

such that the following function obtains its maximum energy:

$$f(\rho, \theta) \stackrel{\triangle}{=} \int_0^1 \int_{-\infty}^{\infty} \psi_0^m(\mathbf{x}, \rho, \theta, t; \sigma) I(\mathbf{x}, t)\, d\mathbf{x}\, dt. \tag{7.8}$$

The necessary condition for this optimization is for the following to be true:

$$\frac{\partial f}{\partial \rho} = \frac{\partial f}{\partial \theta} = 0.$$

Let

$$\psi_1^m(\mathbf{x}, \rho, \theta, t; \sigma) = (3 - \frac{|\mathbf{x}'|^2}{\sigma^2}) \exp\left[-\frac{|\mathbf{x}'|^2}{2\sigma^2}\right]$$

(cf. Eq. (3.4)) and $\mathbf{m}_t = (\cos\theta, \sin\theta)$, $\mathbf{m}_n = (-\sin\theta, \cos\theta)$. The above condition becomes (cf. Eq. (4.5))

$$\frac{\partial f}{\partial \rho} = \int_0^1 \int_{-\infty}^{\infty} \psi_1^m(\mathbf{x}, \rho, \theta, t; \sigma)(\mathbf{x}' \cdot \mathbf{m}_t) I(\mathbf{x}, t)\, d\mathbf{x}\, dt = 0 \tag{7.9}$$

and

$$\frac{\partial f}{\partial \theta} = \int_0^1 \int_{-\infty}^{\infty} \psi_1^m(\mathbf{x}, \rho, \theta, t; \sigma)(\mathbf{x}' \cdot \mathbf{m}_n) I(\mathbf{x}, t)\, d\mathbf{x}\, dt = 0. \tag{7.10}$$

Compared to the spatial receptive field in Chapters 3 and 4, we start with an rf form equivalent to ψ_2 (Eq. (3.4)). This is because the spatio-temporal flow is characterized by elongated volumes rather than by a “zero-crossing” transitions as image contours do.

The above equations are a system with two unknowns. The straightforward solution of it requires partitioning the spatio-temporal space of $\mathbf{x} - t$ using 2D grids of the polar parameterization in (ρ, θ). Instead, we will show that these two variables can be decoupled and, subsequently, we can determine θ independently of ρ. The Taylor expansion of the spatio-temporal image $I(\mathbf{x}, t)$ at $t = 0$ is given by

$$I(\mathbf{x}, t) = I(\mathbf{x}, 0) + \left.\frac{\partial I(\mathbf{x}, t)}{\partial t}\right|_{t=0} t + O(2).$$

Hence, $I(\mathbf{x}, t)$ can be represented, to the first order of approximation, as an algebraic sum of static and dynamic parts. That is,

$$I(\mathbf{x}, t) = I_s(\mathbf{x}) + I_d(\mathbf{x}, t)$$

where $\partial I_s/\partial t = 0$ for all $\mathbf{x}$ and $\partial I_d/\partial t \neq 0$ for all $\mathbf{x}$. Consider the following equation:

$$\int_0^\tau \frac{\partial I(\mathbf{x}, t)}{\partial t}\, dt = I_d(\mathbf{x}, \tau) - I_d(\mathbf{x}, 0) \tag{7.11}$$

Since $I_d(\mathbf{x}, 0)$ is just a reference value, it can can be considered to be zero or, alternatively, we can consider the image to be at rest at $t = 0$. Eq. (7.11) specifies the spatio-temporal volume in which the projected image at $t > 0$ has $\mathbf{m}_t$ field (written as $\mathbf{m}_t(\mathbf{x})$) corresponding to the orientation of optical flow within $t \in [0, \tau]$. The $\mathbf{m}_t$ field can be found using the same method as used for the computation of the tangent field. Given $\mathbf{m}_t(\mathbf{x})$, Eqs. (7.9) and (7.10) become

$$\frac{\partial f}{\partial \rho} = \int_0^\tau \int_{-\infty}^{\infty} \psi_1^m(\mathbf{x}, \rho, \theta(\mathbf{x}), t; \sigma)(\mathbf{x}' \cdot \mathbf{m}_t(\mathbf{x}))I(\mathbf{x}, t)\, d\mathbf{x}\, dt = 0 \tag{7.12}$$

To solve this, we need only a 1D grid for ρ in the $\mathbf{x}$ plane at $t = \tau$ (see Figure 7.2). If an absolute time cannot be defined (or absolute ρ is not relevant), then we can make $\tau = 1$ by normalizing $I(\mathbf{x}, t)$.

7.5 Optical Flow Field Decomposition

For an active observer, optical flow can be very useful if it can be used to infer the relative relationship between the observer and the surface so that further observer motion can be planned. In order to achieve this, we have to relate optical flow to observer motion. This was done in [64] by expressing the canonical fields (i.e., divergence, curl and deformation) in terms of the rotation Ω and translation $\mathbf{v}$ by the observer. In this section, a different formulation is developed, which relates the eigenvalues of these fields to translational observer motion. This alternative form makes the observer motion explicit in order to control the optical flow.

Consider a mobile observer undergoing an instantaneous translation in a static environment. Let $\mathbf{x} = (x, y, z)$ be the coordinates in the observer frame and $\mathbf{q} = (\xi, \eta, 1)$ the projected image coordinates in the 3D Euclidean space. The relative translation velocity is $\mathbf{v} = (v_x, v_y, v_z) = -\partial\mathbf{x}/\partial t$. The observer-centered coordinate system is set up so that the image plane is located at $z = 1$. The object surface can be represented as $(x, y, z(x, y))$. Since $(z_x, z_y, -1)$ is the normal vector to the tangent plane of the object surface at $\mathbf{x}$, we will use $\mathbf{n}$ to denote $(z_x, z_y, -1)$. In this set-up, we have the relationships: $\mathbf{q} = \mathbf{x}/z$, and the optical flow is $\mathbf{u} = \partial\mathbf{q}/\partial t$.

When a rotation $\boldsymbol{\Omega}$ is involved and the transversing translation velocity perpendicular to the line of sight $\hat{\mathbf{x}} = \mathbf{x}/|\mathbf{x}|$ is given by

$$\tilde{\mathbf{v}}_t = \tilde{\mathbf{v}} - (\tilde{\mathbf{v}} \cdot \hat{\mathbf{x}})\hat{\mathbf{x}}$$

the canonical fields can be expressed as (see [64]):

$$\begin{aligned} \nabla \cdot \mathbf{u} &= \mathbf{n} \cdot \tilde{\mathbf{v}}_t + 2\tilde{\mathbf{v}} \cdot \hat{\mathbf{x}} \\ (\nabla \times \mathbf{u})_z &= (\mathbf{n} \times \tilde{\mathbf{v}}_t)_z - 2\boldsymbol{\Omega} \cdot \hat{\mathbf{x}} \\ \zeta_1 - \zeta_2 &= |\mathbf{n} + \hat{\mathbf{e}}_z||\tilde{\mathbf{v}}_t| \end{aligned} \tag{7.13}$$

By definition, $\mathbf{v} = -\partial \mathbf{x}/\partial t$ for a point $\mathbf{x}$ on an object surface, so we can derive

$$\mathbf{u} = \frac{\partial \mathbf{q}}{\partial t} = \frac{1}{z}(-\mathbf{v} + v_z \mathbf{q}) = -\tilde{\mathbf{v}} + \tilde{v}_z \mathbf{q} \tag{7.14}$$

where $\tilde{\mathbf{v}} = \mathbf{v}/z$. By the inverse function theorem it is straightforward to show that

$$\frac{\partial z}{\partial \xi} = -\frac{z_x z}{\mathbf{n} \cdot \mathbf{q}}, \ \frac{\partial z}{\partial \eta} = -\frac{z_y z}{\mathbf{n} \cdot \mathbf{q}}$$

where z_x and z_y are differentials of z with respect to x and y. Using the above formula it can be shown that

$$\begin{pmatrix} \frac{\partial \mu}{\partial \xi} & \frac{\partial \mu}{\partial \eta} \\ \frac{\partial \nu}{\partial \xi} & \frac{\partial \nu}{\partial \eta} \end{pmatrix} = \tilde{v}_z \mathbf{I}_2 + \frac{1}{\mathbf{n} \cdot \mathbf{q}} \begin{pmatrix} \mu z_x & \mu z_y \\ \nu z_x & \nu z_y \end{pmatrix} \tag{7.15}$$

where $\mathbf{I}_2$ is the 2×2 identity matrix.

We can derive the following formulas for the divergence and curl of $\mathbf{u}$:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 2\tilde{v}_z + \frac{\mathbf{n} \cdot \mathbf{u}}{\mathbf{n} \cdot \mathbf{q}} = 3\tilde{v}_z - \frac{\mathbf{n} \cdot \tilde{\mathbf{v}}}{\mathbf{n} \cdot \mathbf{q}} \\ (\nabla \times \mathbf{u})_z &= \frac{(\mathbf{n} \times \mathbf{u})_z}{\mathbf{n} \cdot \mathbf{q}} = \frac{\left[\mathbf{n} \times (\tilde{v}_z \mathbf{q} - \tilde{\mathbf{v}})\right]_z}{\mathbf{n} \cdot \mathbf{q}}. \end{aligned} \tag{7.16}$$

In addition, the symmetric part of $\mathbf{P}$ matrix is given by

$$\mathbf{P}^s = \tilde{v}_z \mathbf{I}_2 + \frac{1}{2\mathbf{n} \cdot \mathbf{q}} \begin{pmatrix} 2\mu z_x & \mu z_y + \nu z_x \\ \mu z_y + \nu z_x & 2\nu z_y \end{pmatrix}.$$

This symmetric matrix can be diagonalized if we choose $\tilde{\mathbf{v}}$ to be such that $\mu z_y + \nu z_x = 0$, i.e., (see Eq. (7.14))

$$\tilde{\mathbf{v}} \cdot (-z_y, -z_x, \xi z_y + \eta z_x) = 0. \tag{7.17}$$

Alternatively we can compute the rotation matrix $\mathbf{Q}$ such that $\mathbf{Q}\mathbf{P}^s\mathbf{Q}^{-1}$ is diagonalized, if the surface shape (z_x, z_y) is already known. The former corresponds to an observer-controlled motion and the latter corresponds to an "off-line" computation. If we diagonalize $\mathbf{P}^s$ by observer motion using Eq. (7.17), it can be shown that the deformation is

$$\zeta_1 - \zeta_2 = \frac{z_x^2 + z_y^2}{\xi z_y + \eta z_x} \frac{(\mathbf{q} \times \tilde{\mathbf{v}})_z}{(\mathbf{n} \cdot \mathbf{q})}.$$

If we diagonalize explicitly by rotation defined by $\mathbf{Q}$, the deformation is

$$\zeta_1 - \zeta_2 = \frac{(z_x^2 + z_y^2)^{1/2}}{\mathbf{n} \cdot \mathbf{q}} (\mu^2 + \nu^2)^{1/2} = \frac{(z_x^2 + z_y^2)^{1/2} |\mathbf{u}|}{\mathbf{n} \cdot \mathbf{q}} \tag{7.18}$$

with the rotation angle γ given by

$$\gamma = \frac{\mu z_y + \nu z_x}{\mu z_x - \nu z_y}.$$

From Eq. (7.16) we can express the optical flow field $\mathbf{u}$ in terms of its curl $(\nabla \times \mathbf{u})$, divergence $(\nabla \cdot \mathbf{u})$ and surface tilt (z_x, z_y):

$$\mathbf{u} = \frac{\mathbf{n} \cdot \mathbf{q}}{z_x^2 + z_y^2} \left[(\nabla \cdot \mathbf{u} - 2\tilde{v}_z)(z_x, z_y) + \nabla \times \mathbf{u}(z_y, z_x) \right] \tag{7.19}$$

and then we can show that

$$\gamma = \frac{1}{z_x^2 - z_y^2} \left[2 z_x z_y + (z_x^2 + z_y^2) \frac{(\nabla \times \mathbf{u})_z}{\nabla \cdot \mathbf{u} - 2\tilde{v}_z} \right].$$

The eigenvalues of the linear vector field defined by Eq. (7.1) (see Section 7.3) is given by

$$2\lambda_{1,2} = 2\left[d \pm (e^2 - c^2)^{1/2}\right]$$
$$= \nabla \cdot \mathbf{u} \pm \left[(\zeta_1 - \zeta_2)^2 - (\nabla \times \mathbf{u})_z^2\right]^{1/2}.$$

From Eq. (7.16) and Eq. (7.18) it can be shown that $(e^2 - c^2)^{1/2} = d - \tilde{v}_z$. Hence the two eigenvalues are given by

$$\lambda_{1,2} = \tilde{v}_z,\ \ \tilde{v}_z + \frac{\mathbf{n} \cdot \mathbf{u}}{\mathbf{n} \cdot \mathbf{q}}. \tag{7.20}$$

An alternative way of deriving this result is by directly solving Eq. (7.6). From this form, we can see that the eigenvalues are always real, which is consistent with the elimination of the curl field in the first place, since it is not useful in solving for scene geometry [64]. Without a curl field, the matrix **P** is symmetric and the deformation is given by

$$|\lambda_1 - \lambda_2| = \left|\frac{\mathbf{n} \cdot \mathbf{u}}{\mathbf{n} \cdot \mathbf{q}}\right|. \tag{7.21}$$

If we assume that the canonical fields can be observed and computed by the observer, Eq. (7.21) allows us to determine the surface normal without having to compute the divergence field. In essence, the deformation field tells us the surface orientation.

7.6 Segmentation of the Optical Flow Field

One of the purposes of optical flow segmentation is to identify discontinuity of surface geometry. The occurrence of discontinuity is due to either the presence of discontinuous contours on an object surface or to discontinuity in depth. Though there is no unique interpretation of the results from segmentation, the result does strongly constrain the problem of identifying object boundaries.

For an observer-controlled translational motion $\mathbf{v} = (v_x, v_y, v_z)$, the optical flow field $\mathbf{u}$ is given by (Eq. (7.14))

$$\mathbf{u} = -\tilde{\mathbf{v}} + \tilde{v}_z \mathbf{q}.$$

To measure the smooth characteristics of the optical flow, two factors have to be considered: the orientation and the magnitude of the optical flow. Consequently, a natural definition for the measure of "smoothness" of $\mathbf{u}$, $\epsilon(\mathbf{u})$, is given by the directional derivative of the magnitude of $\mathbf{u}$ in the direction of $\mathbf{u}$, i.e.,

$$\epsilon(\mathbf{u}) \triangleq \nabla(|\mathbf{u}|) \cdot \frac{\mathbf{u}}{|\mathbf{u}|}. \tag{7.22}$$

Since

$$|\mathbf{u}| = \left(\mu^2 + \nu^2\right)^{1/2} = \frac{1}{z}\left[(-v_x + \xi v_z)^2 + (-v_y + \eta v_z)^2\right]^{1/2} \triangleq \frac{A^{1/2}}{z}$$

it follows that

$$\begin{aligned}
\nabla(|\mathbf{u}|) \cdot \frac{\mathbf{u}}{|\mathbf{u}|} &= \frac{z}{A^{1/2}}\left[-\frac{A^{1/2}}{z^2}(\mu z_x + \nu z_y) + v_z A^{-1/2}\left(\mu^2 \frac{\partial \xi}{\partial x} + \mu\nu \frac{\partial \eta}{\partial x} + \mu\nu \frac{\partial \xi}{\partial y} + \nu^2 \frac{\partial \eta}{\partial y}\right)\right] \\
&= -\frac{\mathbf{u} \cdot \nabla z}{z} + \frac{z v_z}{A}\mathbf{u} \cdot (\mu \nabla \xi + \nu \nabla \eta)
\end{aligned}$$

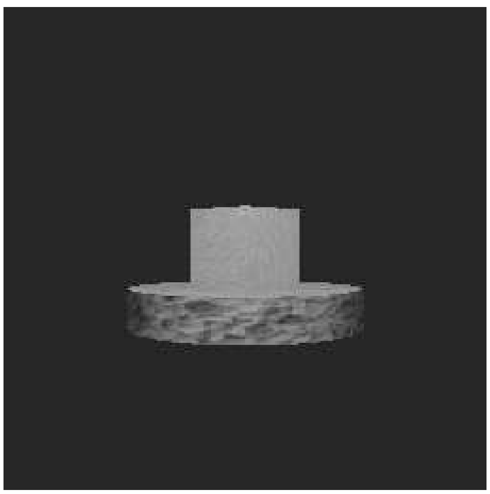
Simplify the expression and it can be shown that

$$\epsilon(\mathbf{u}) = \frac{1}{z}\left[-(\mathbf{n} \cdot \mathbf{u}) + \tilde{v}_z\left(1 - \frac{(\mathbf{q} \cdot \mathbf{u})(\mathbf{n} \cdot \mathbf{u})}{|\mathbf{u}|^2}\right)\right] \tag{7.23}$$

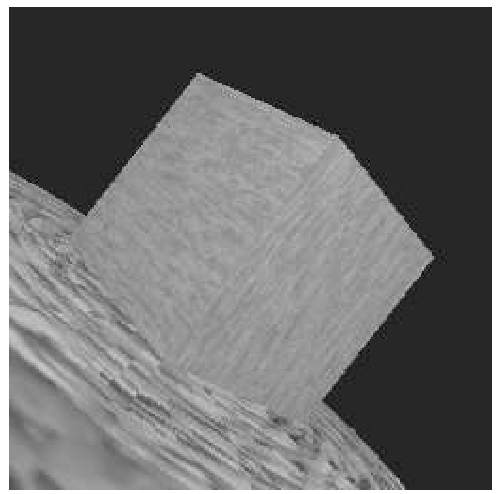
If we treat $\epsilon(\mathbf{u}(\xi, \eta)) = \epsilon(\xi, \eta)$ as a function defined on image plane, changes (i.e., discontinuities) in the structure of ϵ will correspond to image contours. The boundary where these changes occur is a consequence of changes in scene geometry, which involves surface normal $\mathbf{n}$ and depth z as reflected in Eq. (7.23).

7.7 Examples

The first test sequence[1] to be used is shown in Figure 7.3. The spatio-temporal volume of the sequence is shown in Figure 7.4. This is a sequence of 20 images with both curl and divergence fields. The dynamic or differential image, $I_d(\mathbf{x}, t)$, of the spatio-temporal volume is shown in Figure 7.5. The integral curves of the flow field with respect to time are made apparent by smoothing the local image structure (Figure 7.6).



(a) First frame of an image sequence.



(b) Last frame of an image sequence.

Figure 7.3: First test image sequence.

For optical flow, a window of 5 frames was used to compute the integral curve for the flow field. These curves were computed at four scales: 1.5, 2, 3, and 4. The orientation of the optical flow at each point was computed using the method in Chapter 4, and the result for 10th frame is shown in Figure 7.7. The magnitude of the flow field was computed using Eq. (7.12), with a grid of $\rho = 1, 2, 3, 4, 5$, followed by linear interpolation to locate the zero-crossing point. The components of the curl and divergence fields are clear in the spiral shape of the optical flow. The scale which was used to identify the magnitude of the optical flow is not shown in

[1] SOFA synthetic sequences, courtesy of the Computer Vision Group, Heriot-Watt University

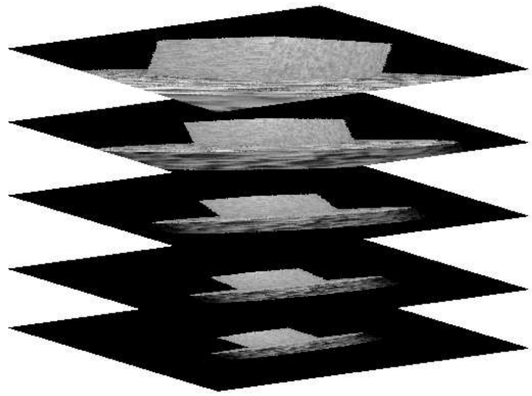


Figure 7.4: Spatio-temporal volume of the image sequence (5 of 20 images).

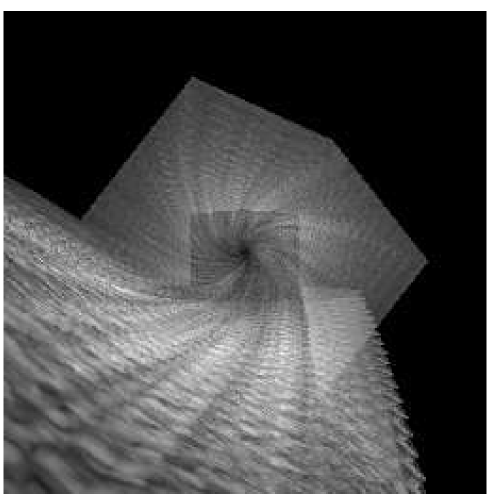


Figure 7.5: The differential image computed by $I_d(\mathbf{x}, \tau) - I_d(\mathbf{x}, 0)$.

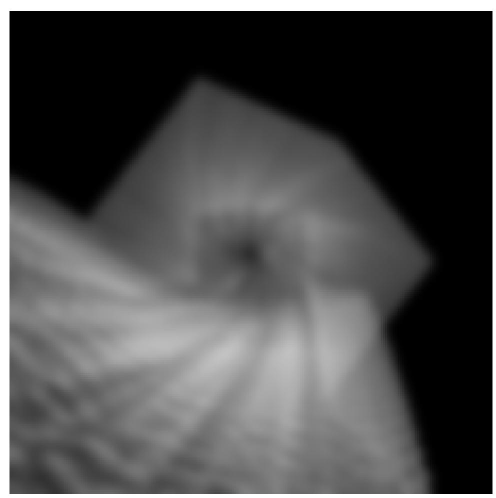


Figure 7.6: The differential image of Figure 7.5 after Gaussian smoothing.

the results, but it is an indication of the nature of local texture.

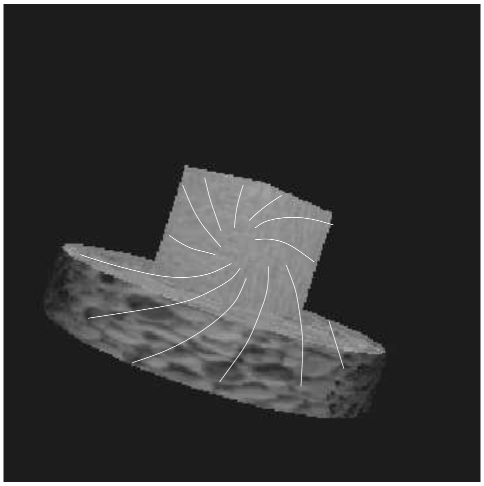


Figure 7.7: The integral curve of the optical flow field for frame 10.

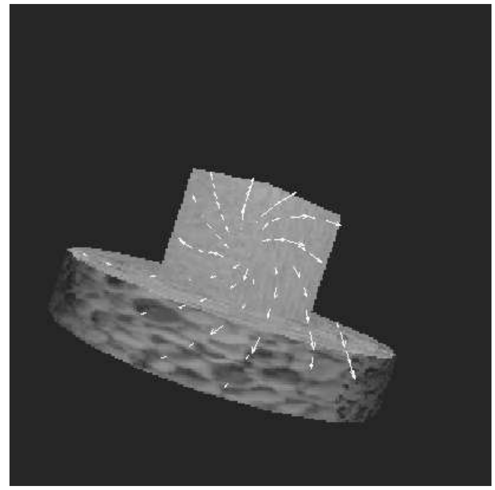


Figure 7.8: The optical flow field computed for frame 10.

The second test sequence[2] is shown in Figure 7.9. This is a synthetic sequence of 15 frames that shows a fly-through of Yosemite valley. The segmentation method formulated in Eq. (7.22) was applied to the optical flow of the eighth frame of the sequence (Figure 7.10). The magnitude ($|\mathbf{u}|$) and the orientation ($\mathbf{u}/|\mathbf{u}|$) parts are represented by gray-level images in Figures 7.11 and 7.12, respectively. The measure of directional derivative, $\epsilon(\mathbf{u})$, is shown in Figure 7.13, encoded and equalized for gray-level representation. By combining the measure of segmentation and the cues provided by the magnitude and orientation of the optical flow, the boundaries of objects are shown in Figure 7.14.

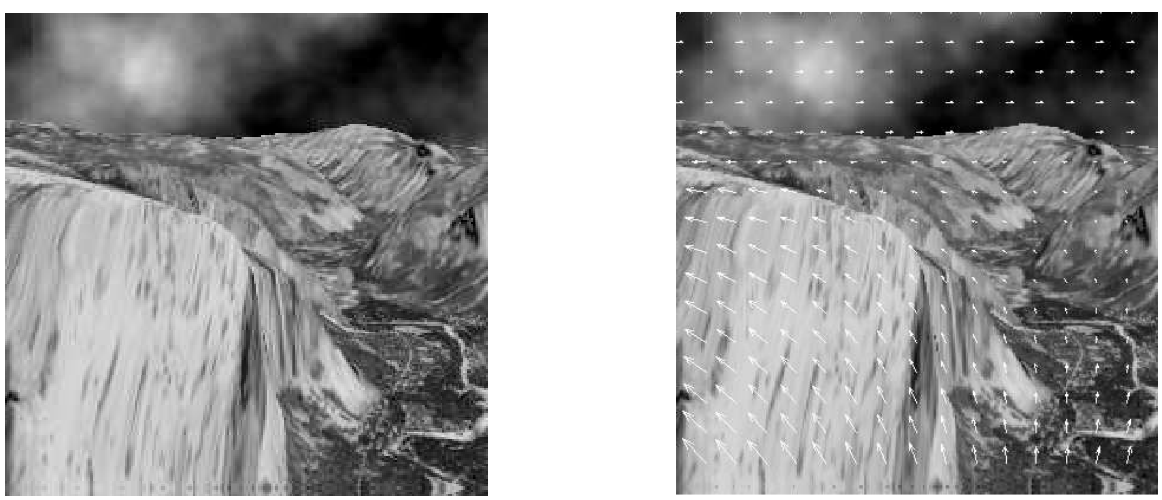
7.8 Summary

The major results presented in this chapter are: (1) a computational mechanism for computing optical flow using polar parameterization, and (2) a new method for optical flow segmentation. The local kernel for computing optical flow is consistent with the geometric operations

[2]The Yosemite Fly-Through sequence produced by Lynn Quam at SRI.

(a) First frame of the image sequence. (b) Last frame of the image sequence.

Figure 7.9: Second test image sequence.



(a) Frame 8. (b) Optical flow for frame 8.

Figure 7.10: Frame 8 and its optical flow.

Figure 7.11: The magnitude of the optical flow for frame 8.

Figure 7.12: The orientation of the optical flow for frame 8.

Figure 7.13: The gray-level representation of the segmentation of the optical flow.

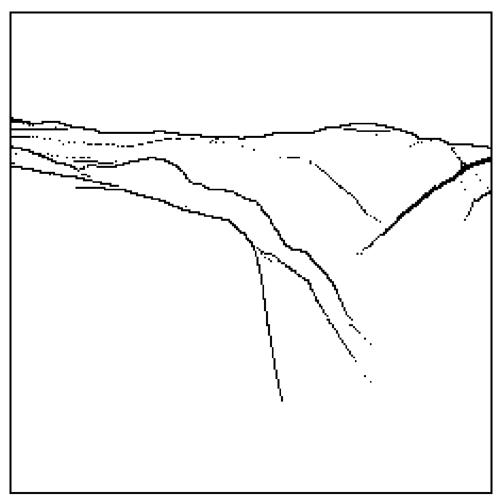


Figure 7.14: The binary segmentation of the optical flow.

presented in previous chapters in both its mathematical form and its interpretation in terms of how biological systems implement similar functionality. In addition, for an active observer, its motion can be used to control the shape of the decomposed vector field of optical flow. This property enables the observer to determine its orientation with respect to the object surface or navigate around the object.

The result from the segmentation of optical flow provides the observer with strong hypotheses regarding the type and location of object boundaries. This information can be effectively employed when the observer is active and can control its motion. These aspects will be examined in the next chapter.

Chapter 8

Global Surface Representation and Navigation

For computational vision, the global representation of 3D objects is both a perceptual and a mathematical problem. These two aspects complement each other and have to be studied close to each other, since our goal is both to understand the human visual system and to design compatible artificial systems. The key issue regarding the computation of surface features and the use of them in the representation of surface shape is to identify perceptually meaningful properties that can be efficiently computed and effectively characterize the global surface shape.

Developing a theory for surface representation from given features is similar to studying the equilibrium of a physical system, in which all objects are at equilibrium with each other. On the other hand, to decide which representation to use and how the elements in a representation can be computed is analogous to the study of the dynamical process before a physical system reaches equilibrium. These two computational aspects of a 3D problem need to be studied in conjunction with the mathematical and perceptual issues for a complete understanding of vision.

In this chapter, the 3D representation problem is studied to provide a mathematical framework for the perceptual process, and the computation of representations is investigated using methods of active navigation.

Surface Representation Instead of using a predefined method of 3D surface representation (such as those used in computer graphics), a method is used here that ties perception to its mathematical representation so that geometric "features" directly correspond to the results of visual perception, as examined by psychology and psychophysics. Since perception itself is constrained by the resources available (e.g., time available to examine an object) and the task to be performed (e.g., to avoid collision or to examine an artifact), the precision and exact form of representation is not pre-determined, i.e., it is shown that 3D surfaces can be represented by features that are (1) perceptually meaningful (namely, derived from geometric features), (2) can be computed efficiently (through local computation), and (3) have global significance for the surface shape representation. This representation scheme is studied in the first part of this chapter.

Surface Navigation When a 3D scene is projected onto the image plane and forms 2D images, there is a formal, though ill-posed, connection between the scene and the images. An active observer can carry out controlled motions to recover this connection and establish the scene geometry from the 2D projected images collected along the navigation path. This motion may be used as part of a plan to collect additional information that is not available from current observation, or as a way to verify hypotheses formed from current observation. The methods developed in previous chapters for inferring surface shape from textured surfaces and optical flow provide strong hypotheses about the surface shape. Consequently, navigation becomes essential in verifying or falsifying these hypotheses. This is also a natural way of integrating methods for the problem of surface recovery by an active observer.

In the second part of this chapter, we present methods for surface navigation using contour information given that appropriate reference frames can be obtained, either from outside the surface (extrinsic) or within the surface (intrinsic).

8.1 Global Features on the Surface

Features, by their very nature, are properties of a region of the surface that present themselves to the sensor in a way that is perceptually different from neighboring regions. It is also desirable to have a well-defined correspondence between the feature and the region from which it is computed. In other words, features are both perceptually distinguishable and mathematically representative (in terms of surface geometry). Here, surface shape is considered in the sense of differential geometry and the representation of the surface is a result of extending the shape represented by a feature to the whole region of the surface from which the feature is computed. In this sense, the most fundamental geometric features are points on the surface that exhibit special geometric properties.

According to proposals by Attneave [8] and Koenderink [61, 65], perceptually, local curvature is the focus of feature computation. Consequently, a point feature is a point within a region on the surface where a second-order geometric measure attains its extremal value at this point in relation to its neighborhood. From this geometric view, if the surface curvature at every point can be determined or estimated, all features can be computed. In previous chapters, a series of methods were already developed for the computation of local geometry on a surface. As a consequence, we can assume here that, given a marked point on an object surface satisfying certain "visibility" criteria, the position and local shape can be computed directly or can be estimated by appropriately changing vantage point.

Let's start with the definition of the formal concepts of feature and related properties. First of all, we need to have an idea of what can be observed and what can not, since all the computation is based on observation and what cannot be seen cannot be computed. Corresponding to static and apparent contours on the surface, the visibility of surfaces also comes in two varieties: visible and "apparently visible."

Definition 8.1.1 (Visibility). *For an observer, a point is* visible *on the surface if there is a ray* *emanating from the point that does not intersect with any other part of the surface. A point on the surface is* apparently visible *if there is a ray tangent to the surface which emanates from the point that does not intersect the surface.*

Definition 8.1.2 (Visible Surface). *The visible part of a surface is the collection of all the visible points on the surface and the apparently visible part of a surface is the collection of all the apparently visible points on the surface.*

In differential geometry, the surface shape at a given point is defined by the *normal curvature* of a parametric curve passing through the given point. It is also convenient to define a feature point using surface curves.

Definition 8.1.3 (Characteristic Path). *A surface curve passing through a given point on a surface is a* characteristic path *for this point if the curvature is an extremum at the point.*

Definition 8.1.4 (Feature Point). *A point on a smooth surface is a* feature point *if it is a static surface mark or it has a characteristic path.*

Definition 8.1.5 (Prominent Feature Point). *A* prominent *feature point is a point where every path passing through the point is a characteristic path for the point, with possibly a finite number of paths of constant curvature.*

Definition 8.1.6 (Feature Curve). *A 3D curve is a* feature curve *(or curvilinear feature) if every point on the curve is a feature point or if the curve is part of a discontinuous contour.*

With these definitions, we can begin the study of the representation of surfaces under the condition that a set of feature points is already given. The second part of the chapter will show how an active observer can navigate around the surface and locate the features defined above.

8.2 Global Surface Shape Representation

The local geometry of 3D surfaces is specified by the two fundamental forms in differential geometry. This local description is smooth differentially and can describe arbitrary smooth surfaces. However, the descriptive power requires infinite resources and cannot be used as the basis for tasks such as object recognition. In this section, methods are developed that embody the idea that a local extremum point in curvature space can be extended to be representative of a global region in the neighborhood of this feature point.

The basic method employed here is interpolation of surface curves. In contrast with interpolation of only the location of points, we start with surface points that are described in fully differential geometry language, i.e., two differential fundamental forms. It is shown that this geometric description generates natural surface curves that can be used to strongly constrain the surface shape and, subsequently, the curves can be extended from a local shape representation to a global one. The key is to keep the shape unchanged throughout its neighborhood or changed smoothly to the next description point along an interpolated planar path.

The scheme is presented in an organic way by first introducing how a single feature point can be representative of a region and its underlying volume. This is followed by two point representations of the surface. Since the surface can always be triangulated when there are more than three points, the final step is to show how a triangular patch of the surface can be constructed from three feature points while leaving the local shape around the feature points intact. This method of surface representation is an example of *incremental modeling*, in which each additional piece of shape information contributes incrementally to the surface model without imposing global changes to the original model.

8.2.1 Surfaces From a Single Point

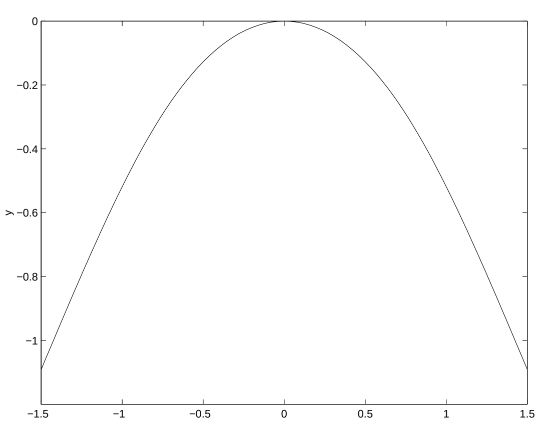
A point feature is prominent if it is a feature point for any planar curve passing through the point, i.e., each planar curve has a monotonically decreasing curvature magnitude. We can use a generic monotonic function such as a Gaussian to represent the curvature of such a curve. However, the Gaussian function is not adequate for the curve itself because it has three feature points rather than one (Figures 8.2 and 8.3). Instead, the Gaussian will be used to represent the curvature of the destination curve using Eq. (3.15), i.e.,

$$\mathbf{c}(s) = \left(\int \cos(\frac{\sqrt{\pi}}{2} \operatorname{erf}(s))\, ds, \int \sin(\frac{\sqrt{\pi}}{2} \operatorname{erf}(s))\, ds \right) \tag{8.1}$$

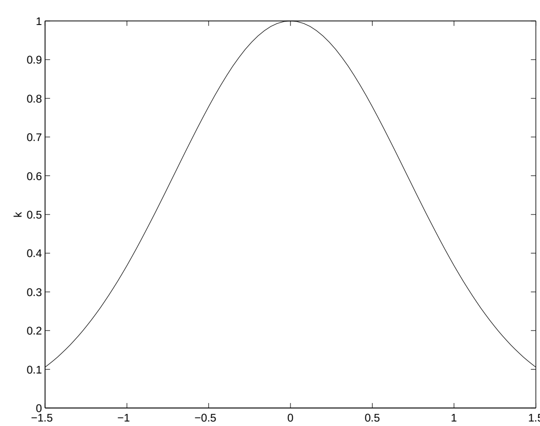
since

$$\int \exp(-s^2)\, ds = \frac{\sqrt{\pi}}{2} \operatorname{erf}(s)$$

where $\operatorname{erf}(s)$ is the *error function*. The prototypical curve with unit curvature is shown in Figure 8.1.



(a) A curve



(b) The curvature

Figure 8.1: A curve with Gaussian curvature profile.

Given a smooth surface, the process of constructing another surface with local shape conforming to the shape of the given one is to generate two planar curves that will become curvature lines for the surface (Figure 8.4). Hence the surface resulting from a prominent feature point is prominent geometrically in the local region and the part of the surface in the immediate neighborhood is completely characterized by the two principal curvatures. This surface shape is uniquely defined by the two prototype curves in Figure 8.1. Both an elliptic and a hyperbolic surface specified in this way are shown in Figure 8.5.

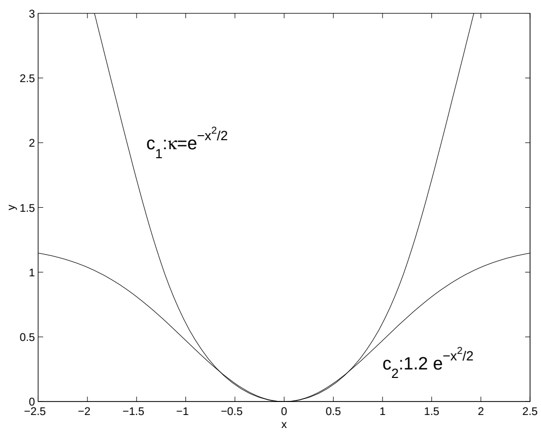


Figure 8.2: Gaussian curve and the curve with Gaussian curvature.

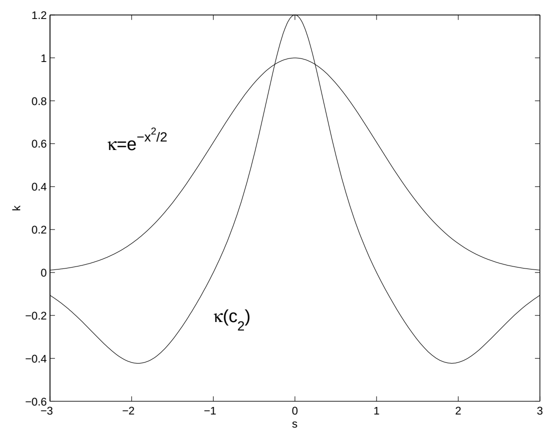


Figure 8.3: Gaussian curvature and curvature of a Gaussian curve.

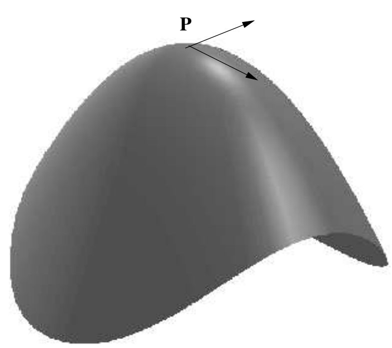
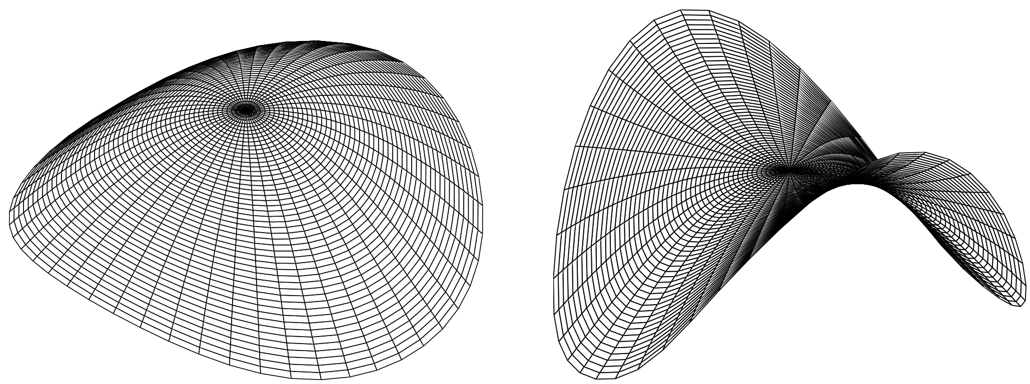


Figure 8.4: Surface shape extension from a single point.



(a) An elliptic surface ($\kappa = 2, 1$) **(b) A hyperbolic surface ($\kappa = 2, -1$)**

Figure 8.5: Surfaces with elliptic and hyperbolic curvatures.

8.2.2 Surfaces From Two Points

When two surface curves are defined, one of the natural surface interpolation schemes is the *tensor-product surface* [56]. However, to maintain the shape of two surface feature points, three curves need to be defined from two single point extensions and a planar curve interpolation between these two points (Figure 8.6) and therefore the tensor-product surface method is not applicable. In this section, another method is developed for this case.

Assume we are given a surface S:$\mathbf{x}(u, v)$ parameterized by (u, v) and three distinct curves on the surface: two kernel curves C_1, C_2 given by $\mathbf{x}(u, 0)$ and $\mathbf{x}(u, v_2)$, respectively, and a path curve C_p:$\mathbf{x}(0, v)$. We want to formalize the concept that the surface can be considered as the result of moving C_1 along C_p while at the same time gradually deforming it until it reaches v_2 where it eventually becomes C_2. We will consider the case where the deformed curve is transformed from C_1 by a linear transformation. This is pertinent in our case since both C_1 and C_2 are locally extended from a single surface point using the same principle (see previous section).

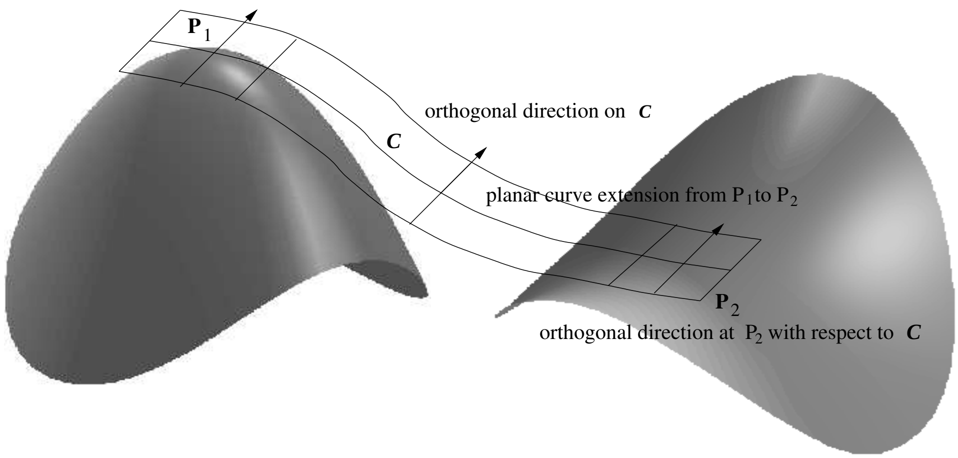


Figure 8.6: Surface shape extension from two surface points.

Let the transformation rule be

$$\mathbf{x}(u, v) = \mathcal{L}(\mathbf{x}(u, 0), v) \stackrel{\triangle}{=} \mathcal{L}_s(v).$$

Since $\mathcal{L}_0(v)$ maps to C_p (i.e., $\mathbf{x}(0, v)$), we have

$$\mathbf{x}(u, v) = (\mathbf{x}(u, 0) - \mathbf{x}(0, 0))\mathbf{L}(v) + \mathbf{x}(0, v) \tag{8.2}$$

where $\mathbf{L}(v)$ is a linear transformation, with $\mathbf{L}(0) = \mathbf{I}$. For a given v, C_k and $\mathbf{x}(u, v)$ are parallel and there is a constant vector $\mathbf{p}$ such that

$$\mathbf{p} \cdot \mathbf{x}(u, v) = b(v) \tag{8.3}$$

where $b(v)$ is a scalar function of t. From Eq. (8.2) and Eq. (8.3), we have

$$\mathbf{p} \cdot (\mathbf{x}(u, 0)\mathbf{L}(v)) = a(v) \tag{8.4}$$

where $a(v) = b(v) + \mathbf{p} \cdot (\mathbf{x}(0, 0)\mathbf{L}(v) - \mathbf{x}(0, v))$. Since

$$\mathbf{p} \cdot (\mathbf{x}(u, 0)\mathbf{L}(v)) = (\mathbf{p}\mathbf{L}^T(v)) \cdot \mathbf{x}(u, 0)$$

Eq. (8.4) becomes

$$(\mathbf{p}\mathbf{L}^T(v)) \cdot \mathbf{x}(u, 0) = a(v).$$

Since this is the plane occupied by $\mathbf{x}(u, v)$ and is parallel to the plane with normal $\mathbf{p}$, the following condition must be satisfied:

$$\mathbf{p}\mathbf{L}^T(v) = \lambda(v)\mathbf{p}.$$

That is, the normal vector, $\mathbf{p}$, of the plane where C_k resides is the left eigenvector of the linear transformation $\mathbf{L}^T$. Let the eigenvalues be $\lambda_i(v)$, $i = 1, 2, 3$, and the corresponding eigenvectors be $\mathbf{p}_i$. If matrix $\mathbf{S} = [\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3]$ we have

$$\mathbf{L}^T(v) = \mathbf{S}^{-1} \begin{pmatrix} \lambda_1(v) & 0 & 0 \\ 0 & \lambda_2(v) & 0 \\ 0 & 0 & \lambda_3(v) \end{pmatrix} \mathbf{S}$$

Since $\mathbf{L}(0) = \mathbf{I}$, it follows that $\lambda_i(0) = 1$.

Given shape descriptions for two points, P_1, P_2, (complete fundamental form description) on a surface, the planar curve C_p is given by the curve connecting P_1 and P_2 on the plane E_p determined by P_1, P_2 and the surface normal vector $\mathbf{n}_1$ at P_1. The planar curve C_k is defined by the surface curve C_1 passing through P_1 and on a plane E_k orthogonal to the plane E_p. Furthermore, for a given parameterization v of C_p, the planar curve C_2 passing through P_2 and orthogonal to E_p is constrained to be C_2:$\mathbf{x}(u, v_2)$ (Figure 8.6). This latter constraint can be used to determine the eigenvalues since $\mathbf{x}(u, v_2) = (\mathbf{x}(u, 0) - \mathbf{x}(0, 0))\mathbf{L}(v_2) + \mathbf{x}(0, v_2)$. Let

$\mathbf{x}(u, v) = (x_1(u, v), x_2(u, v), x_3(u, v))$. It can easily be shown that if C_2 is *similar* to C_1 in the sense that for $i = 1, 2, 3$,

$$x_i(u, v_2) - x_i(0, v_2) = (x_i(u, 0) - x_i(0, 0)) f_i(v_2)$$

$\mathbf{L}(v)$ will be diagonal with $\lambda_i(v) = f_i(v)$. The similarity condition formalizes the notion that the deformation of C_k along C_p should be uniform along the u parameterization and, hence, f_i should be independent of u. $\lambda_i(v)$ can be determined by the boundary values

$$\begin{aligned} \lambda_i(0) &= 1, \\ \lambda_i(v_2) &= \frac{x_i(u_0, v_2) - x_i(0, v_2)}{x_i(u_0, 0) - x_i(0, 0)} \end{aligned}$$

This similarity condition is guaranteed by the local extension of P_1, P_2 in the u direction and we can choose, e.g., $u_0 = 1$ to derive $\lambda_i(v_2)$, and Hermite functions can be used to interpolate these two endpoints:

$$\lambda_i(v) = H_{0,0}(\frac{v}{v_2}) + H_{0,1}(\frac{v}{v_2})\lambda_i(v_2)$$

From Eq. (8.2) our surface extended from the two points P_1, P_2 becomes

$$\mathbf{x}(u, v) = (\mathbf{x}(u, 0) - \mathbf{x}(0, 0)) \begin{pmatrix} \lambda_1(v) & 0 & 0 \\ 0 & \lambda_2(v) & 0 \\ 0 & 0 & \lambda_3(v) \end{pmatrix} + \mathbf{x}(0, v) \qquad (8.5)$$

Since the plane E_k is orthogonal to the plane E_p we can alway choose, for example, E_k to be the $x-z$ plane and E_p be the $y-z$ plane in Cartesian coordinates. In this case, we can immediately see from Eq. (8.5) that $\lambda_2(v) = 1$, and only $\lambda_1(v)$ and $\lambda_3(v)$ need to be determined.

In Figure 8.7 the surface extended from two known points P_1, P_2 is given, with P_1 being an elliptic (Gaussian curvature $K > 0$) point and P_2 being a hyperbolic point ($K < 0$). The C_p

curve is given in Figure 8.10. The surfaces with both points elliptic and both hyperbolic but the same C_p, are given in Figures 8.8 and 8.9, respectively.

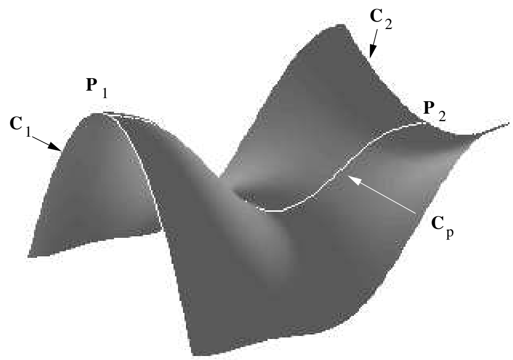


Figure 8.7: Surface shape extension from one elliptic and one hyperbolic point.

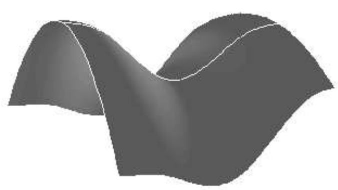


Figure 8.8: Surface shape extension from two elliptic points with positive curvature.

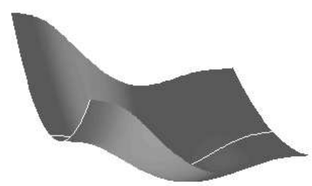


Figure 8.9: Surface shape extension from two elliptic points with negative curvature.

8.2.3 Surface From Multiple Points

When there are three or more feature points in a region with known local shape, a triangular mesh is formed and the surface can be constructed by patching the part whose shape is not specified by the feature points. Given three disparate feature points and their associated surface shape, we have inherited three patches as a constraint. To construct a surface patch connecting these three patches under the constraint, an area will have to be delimited around each feature point so that the shape can be kept intact. The triangular region defined by connecting the three

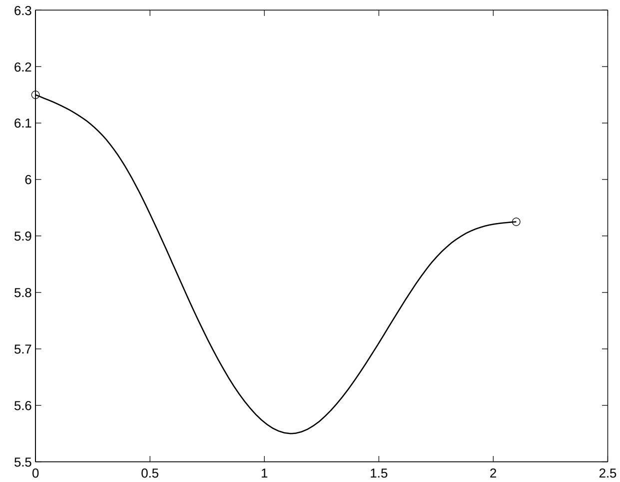


Figure 8.10: The C_p curve.

points while excluding the reserved areas forms a six-sided patch. Hence the problem becomes the construction of a six-sided patch with given boundary curves. One method for solving this problem is the Gordon-Coons surface patch.

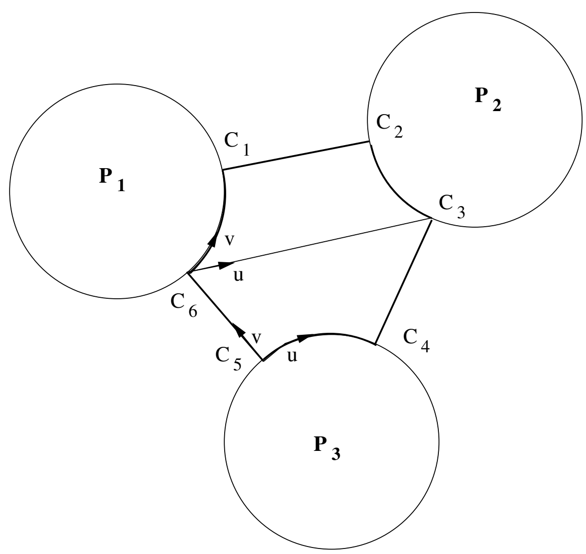


Figure 8.11: The construction of a six-sided Coons patch.

When given four boundary curves $\mathbf{x}(u, 0)$, $\mathbf{x}(u, 1)$, $\mathbf{x}(0, v)$, $\mathbf{x}(1, v)$ on the surface and their tangent derivatives along the curves, a patch with continuous first-order derivative is given by the Coons patch:

$$\mathbf{x}(u,v) = \mathbf{h}_c(u,v) + \mathbf{h}_c(u,v) - \mathbf{h}_{cd}(u,v)$$

where

$$\mathbf{h}_c(u,v) = H_{0,0}(v)\mathbf{x}(u,0) + H_{0,1}(v)\mathbf{x}(u,1) + H_{1,0}(v)\mathbf{x}_v(u,0) + H_{1,1}(v)\mathbf{x}_v(u,1), \tag{8.6}$$

$$\mathbf{h}_d(u,v) = H_{0,0}(u)\mathbf{x}(0,v) + H_{0,1}(u)\mathbf{x}(1,v) + H_{1,0}(u)\mathbf{x}_u(0,v) + H_{1,1}(u)\mathbf{x}_u(1,v) \tag{8.7}$$

$$\tag{8.8}$$

and

$$\mathbf{h}_{cd}(u,v) = \begin{pmatrix} H_{0,0}(u) & H_{0,1}(u) & H_{1,0}(u) & H_{1,1}(u) \end{pmatrix} \times \tag{8.9}$$

$$\begin{pmatrix} \mathbf{x}(0,0) & \mathbf{x}(0,1) & \mathbf{x}_v(0,0) & \mathbf{x}_v(0,1) \\ \mathbf{x}(1,0) & \mathbf{x}(1,1) & \mathbf{x}_v(1,0) & \mathbf{x}_v(1,1) \\ \mathbf{x}_u(0,0) & \mathbf{x}_u(0,1) & \mathbf{x}_{uv}(0,0) & \mathbf{x}_{uv}(0,1) \\ \mathbf{x}_u(1,0) & \mathbf{x}_u(1,1) & \mathbf{x}_{uv}(1,0) & \mathbf{x}_{uv}(1,1) \end{pmatrix} \times \begin{pmatrix} H_{0,0}(v) \\ H_{0,1}(v) \\ H_{1,0}(v) \\ H_{1,1}(v) \end{pmatrix} \tag{8.10}$$

The six boundary curves formed by C_1 to C_6 can be partitioned into two rectangular parametric patches by connecting C_3 and C_6, while the tangent vectors at corner points C_i along the curves can either be computed from the patch surface defined by the feature points (e.g., along curve $C_6 - C_1$ from P_1) or interpolated from two patch surfaces (e.g., along curve $C_1 - C_2$ as interpolated from P_1 and P_2). The latter is accomplished by constructing a cutting plane that cuts through C_3 and C_6 (Figure 8.12).

The four cutting planes cut out the two regions that will be connected by the existing feature patches and two yet-to-be-constructed Coons patches (Figure 8.13). The tangents at C_i in the direction along the boundary curves on the three feature patches can be computed directly from Eq. (8.1), while the tangents at C_i along the three cross-patch curves can be interpolated

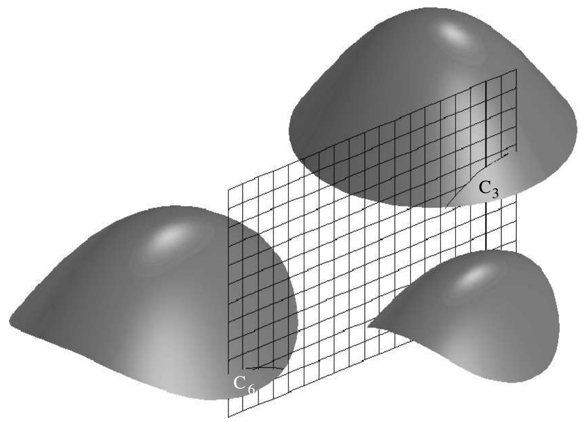


Figure 8.12: The additional cutting plane for Coons patching.

from the two associated cut-out curves (e.g., the cutting plane for $C_1 - C_2$ intersects patch P_1 and P_2 at two cut-out curves associated with $C_1 - C_2$). The problem in joining the two Coons patches together to form a single surface is when crossing the common boundary (i.e., the curve $C_3 - C_6$). If the two Coons patches are computed individually, the tangents along the common boundary will not coincide in general since they are interpolated from different vectors (one from curves $C_6 - C_1$ and $C_2 - C_3$ while the other from curves $C_5 - C_6$ and $C_3 - C_4$). We solve this problem by projecting each interpolated vector $\mathbf{t}_2$ on the second Coons patch to the plane spanned by the interpolated vector $\mathbf{t}_1$ of the first patch and the tangent vector $\mathbf{t}$ on the curve $C_6 - C_3$ (Figure 8.15).

The formulation used in Eq. (8.9) requires both first- and second-order differentials on the boundary curves. We have already discussed how the first-order differentials can be computed directly or interpolated from the given feature patches. The second-order differentials are referred to traditionally as “twist vectors” and can be estimated from the given boundary conditions in order to prevent the Coons surfaces from being “flat.” It is set to zero here to simplify the computational process since the aesthetic appearance of the surface is not a primary concern here.

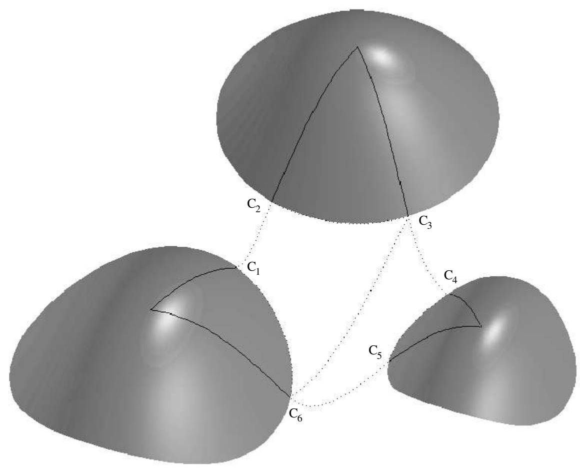


Figure 8.13: The cutting curves for Coons patching.

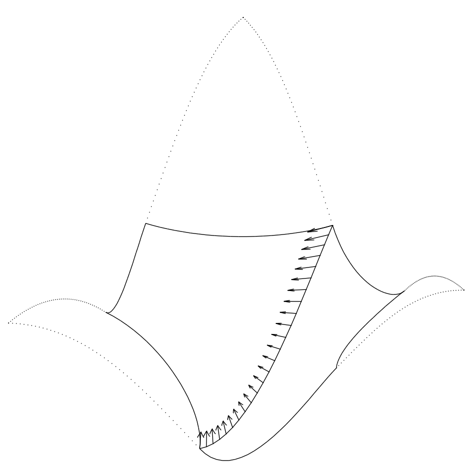


Figure 8.14: The tangent vectors used for Coons patching.

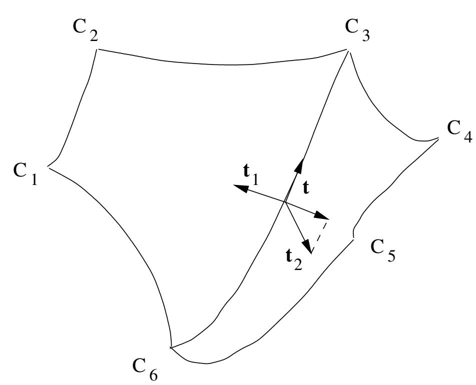


Figure 8.15: Fitting tangent vectors for two Coons patches.

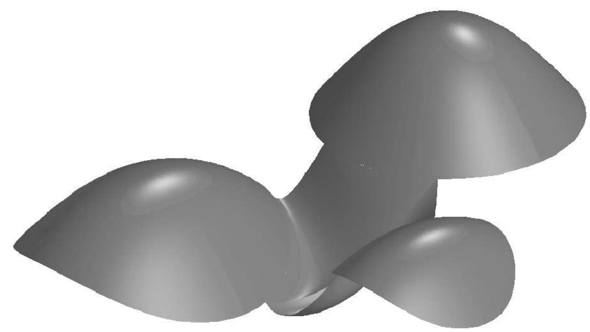


Figure 8.16: The feature patches and Coons patches.

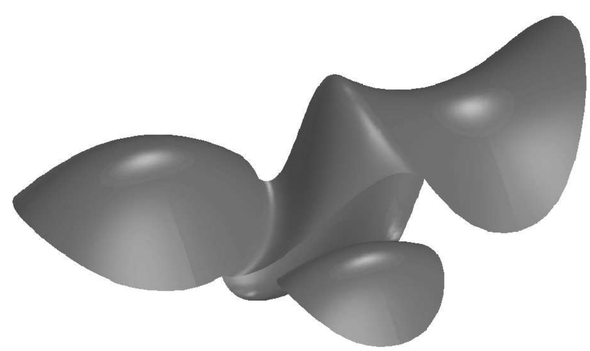


Figure 8.17: Alternative feature patches and Coons patches.

8.2.4 Curves and Structured Features

A surface curve can be perceived as a feature on the surface if it is either a static mark such as a discontinuous contour or a curve whose characteristics are invariant to observer motion. One of the primary uses of these features is to delimit the object surface in an operational way so that either the surface can be partitioned into well-organized subparts or can be recovered from these features by well-defined procedures (e.g., Figure 8.18). One other function these features serve is navigation in which the observer can refer to these static landmarks.

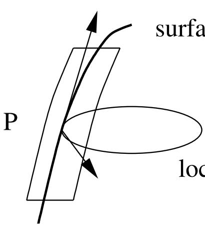


Figure 8.18: Surface shape extension from strip.

As we proceed from primitive features (namely feature points) to higher and more structured features (e.g., curves), the surface is represented in an operationally different way, i.e., different methods are employed to recover the surface, though well-defined conversions can be used to transform between representations. This is essentially a process going from signal processing to information and symbol manipulation. However, the conversion between representations is more of a knowledge acquisition process than inherent in the perceptual system. Alternatively, the conversion can be considered as additional structure imposed on the primitive feature set. Artificial objects with regularity or symmetry are especially prone to be represented by these structured features. Hence the symbolic features serve a primary function of embodying regularity and symmetry or other highly structured organization.

Example 8.2.1. A plane or a sphere has no feature. A single number is used to describe the shape globally.

Example 8.2.2. An ellipsoid has six prominent feature points and three feature curves.

An invariant curve that is not discontinuous will have either all its points being feature points (e.g., a ridge) or part of a symmetric contour (Figure 8.19).

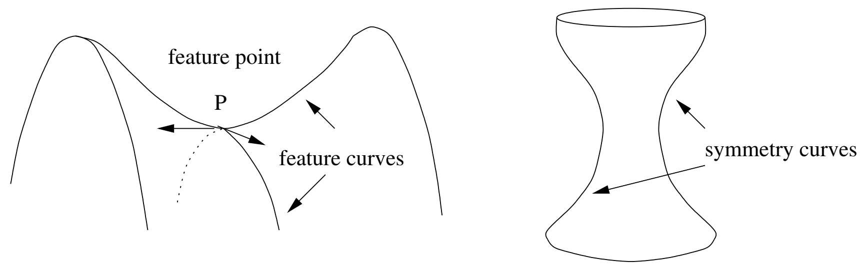


Figure 8.19: Invariant curve as surface features.

In either case, there are well-defined procedures for recovering the surface shape around the curve (see Chapter 6). The shape in the orthogonal direction to the tangent of the curve can be used to constrain the surface and take into account any additional point or curvilinear features available when constructing the surface.

The advantage of using point features to describe global shape is its representational power for individual parts of the surface or the underlying volume as shown in previous sections. However, when there are emerging structures of higher dimensions, these structures become better candidates to organize the representation around these structures (Figure 8.20). This is because of their perceptual properties as stand-alone landmarks intrinsic to the surface. In the following, these structured features will be treated both as aids to navigation and, in cases of symmetry, for representation of surfaces. The difficulty with using higher dimensional features directly in representation is due to the fact that these features have more structure than can be easily handled by the perceptual system. For example, the saddle in Figure 8.19 is much more easily represented by two planar feature curves than as a hyperbolic surface with two asymptotes. In other words, it is easier to identify the features than to represent them. Nonetheless, in the case of planar curves, the representation is essentially a reduced problem of surface representation using feature points, and the effective use of them occurs when there is a natural correspondence between the curve and the volume. As mentioned when point features were discussed, it is the extension of a single feature point to a compact global volume that makes the approach attractive. Similarly, if the representation of a planar curve is all that is needed for representing a volume, it is natural to have the curve as part of the representation tokens. We have already encountered one case in discussing surfaces from two feature points. Another instance is symmetry in shape (Figure 8.21) and will be discussed in the next section.

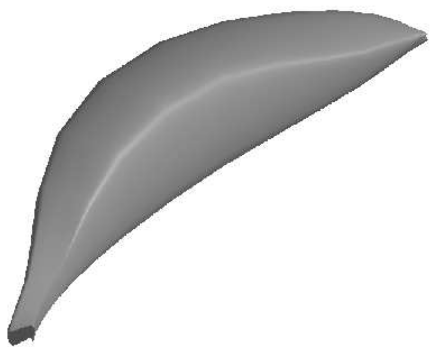


Figure 8.20: Surface shape with feature curves.

Figure 8.21: Surface shape represented by planar curve.

8.3 Global Navigation

The goal of global surface navigation is to locate features that can be used to characterize the surface. These features can be used to represent the surface operationally, i.e., well-defined mathematical methods can be used to recover the surface such that two surfaces with similar features will be considered similar themselves. In other words, an observer will to be able to mark surface features and represent the surface in the region around these features. Hence the navigation procedure involves identifying features, moving in a way so that more information related to the features can be acquired, and representing the parts of surface that have been traversed.

The process of navigation involves the exploration and recovery of "unknown" and "partially unknown" parts of the surface. Unknown parts of the surface are demarcated by discontinuous boundaries. On the other hand, apparent contours, which turn away smoothly from the observer, provide partial information about the surface. Parts of the surface are completely known when they are characterized by a single point that is neither on a discontinuous contour nor on an apparent contour. Hence these three kinds of features are most informative: point, discontinuous curvilinear, and static curvilinear.

The features that leave explicit marks on an object surface are static point marks, static curvilinear features and discontinuous contours. These features are useful in navigating, tracking and representing the surface. On the other hand, apparent contours are useful in identifying geometric feature points such as curvature extrema. Both aspects are goals of navigation.

The shape of the visible part of a surface can be recovered when the surface is textured. This can be accomplished through optical flow or patch computation. The recovery results in the identification of surface features as detailed in previous sections.

8.3.1 Issues

Reference Frame Observer motion comes in two forms: local and global. Local movement conducted by an observer has an observer-centered frame in which the observer moves in a perturbative motion within a local region. Examples of this type of motion include small oscillations and differential rigid motion, which can be expanded into a significant linear term and ignorable higher-order terms using Taylor series. This type of motion can essentially be linearized like all perturbation formulations and henceforth can be easily reversed and tracked [40]. This capability of being able to track where the camera is without resorting to an external reference frame is essential for locating and verifying certain kind of geometric features (see the following sections) as well as acquiring depth cues from optical flow. Non-local movement generally requires an external frame to track the motion and the relative positions between the object and the camera.

Apparent Contours Apparent contours are a powerful but primitive entity in perception. They reveal a great deal about the surface in the direction orthogonal to the viewing direction and the surface normal. However, owing to the lack of effective reference frames, using local shape information from apparent contours to assemble the global shape of a surface is inefficient and difficult (Figure 1.4). This approach can work well when an adequate reference frame is provided, such as the scanning a solid model placed on a rotating platform. Hence the primary usefulness of apparent contours is for identifying plausible features as hypothesized by, for example, texture tracking or optical flow.

Static Surface Features Point marks and stationary curvilinear contours are embedded features on an object surface, which do not motivate navigational motion themselves. This is because there is no apparent or dictated movement that will assure information gain. However, these features convey surface shape information (Chapter 6), which can be useful in representing global shape from a fixed vantage point, and serve as tracking and representation references. On the other hand, a discontinuous contour itself already imparts a great deal of information to the observer and, additionally, suggests motion paths for maximum information gain.

The mathematical formulation of discontinuous contours is somewhat different from how they are perceived. Hence we will characterize the "discontinuity" by the change of curvature over a small spatial region, which is scale-dependent. Furthermore, discontinuous contours deform because of projective geometry. This is qualitatively different from apparent contours that deform because of surface geometry, in addition to the projective deformation. However, there are observer movements that can eliminate the projective deformation, which will be discussed in the next section.

For an active observer in motion, an apparent contour manifests its deformation through a convex trajectory orthogonal to the contour along the surface. A static contour also deforms as an observer moves, but in a different way (see Chapter 6). This difference of deformation can only be identified if we can compute the surface shape by means other than the apparent contour.

Scales Scales, such as σ in the family of receptive fields, are key parameters of the sampling process in visual perception, which preserve all the continuous properties essential for computation. The hierarchy of the physical events embedded in scale space provides a means to control the distribution of those events while maintaining the ability to examine any event in any level of the hierarchy in a well-defined way. This is essential in both computing surface shape from static surface marks and apparent contours.

The effects and advantages of using a scale-space structure to process two-dimensional images were already discussed in Chapters 2 and 3. These scale-space related operations are an integrated part of the visual front-end, which directly samples the optical input. The surface shape, in turn, is computed from the projection of the optical input (see Section 3.3.1.1). Mathematically, the scale-space operation is a blurring operation and will affect the number of points being tracked but not their projections onto the image plane. This is because adjacent points will merge into fewer ones or disperse into oblivion. In effect, this make the enclosing surface more "flat" visually since the average distance between point features increases as the scale gets coarser.

The scale effect is always part of the two-dimensional imaging process. However, there is a correlation between the two-dimensional scale space and three-dimensional surface recovery, which will become clear next.

8.3.2 Formal Properties of Features

For the purpose of identifying features for surface representation, the observer needs to evaluate what can be computed from the current viewpoint, what is a reasonable hypothesis for the unknown part of the surface, and determine how to move based on this information. Some properties of features are independent of the surface geometry and other properties are useful for the observer to plan its motion. In this section, we will study these properties.

Within a region without discontinuous contours, the surface is smooth and can be represented hierarchically in the framework of scale space. In the case when the surface is textured, the initial shape of the surface can be computed from triangulated surface patches. According to Definition 8.1.4, we can compute all the feature points for a visible surface region, given a fixed scale and vantage point. The boundary of the region is composed of either discontinuous or occluding contours. A discontinuous contour is itself a feature (Definition 8.1.6). For an occluding contour, all the feature points on the contour are either prominent feature points or part of curvilinear features or both. This property can be used for two purposes: to guide global navigation and to identify local surface shape. Global navigation is a logical consequence of the effort to recover surface shape around the feature points on the contour through the observer's movements. Two types of movement are among the most effective ones: navigation using landmarks, and movement by localized motion (perturbative motion). Two kinds of landmarks are useful for navigation and surface recovery: external reference frame and surface features, which include both static surface marks and prominent feature points. A prominent feature point on an apparent contour can be identified by local perturbative motion, since only three observations are needed to uniquely recover the shape around the point (see Eq. (3.25)).

8.3.2.1 Scale

Given a scale, all the prominent feature points are separated by at least a distance determined by that scale when they are projected onto the image plane. This entails a finite set of prominent feature points on the surface, and scale space can be used to control the size of the finite set. Part of the goal of surface recovery is to locate this finite set of features for the given scale. The above property is captured in the following two formal statements.

Proposition 8.3.1. Given a function $f(x)$ with bounded second-order differentiation, there is a scale such that the distance between any two adjacent zero-crossing points of the function is lower-bounded by the distance, if a fixed threshold δ is set as the zero-crossing criterion.

From the above proposition, it follows that any finite region in the image plane can only have a finite number of zero-crossing points for any smooth and bounded 2D function. Hence we have the following proposition.

Proposition 8.3.2. For any bounded region of a surface, the projection of the region onto the image plane has only a finite number of prominent feature points.

8.3.2.2 Apparent Contour

Apparent contours are one of the primary sources for navigation. Some of its formal properties are studied here, which will be useful later.

The following proposition follows directly from the definitions.

Proposition 8.3.3. The apparently visible part of a surface is a subset of the visible part of the surface.

A prominent feature point can be characterized by the language of differential geometry by the following theorem, which will become useful when an observer navigates to recover the surface shape.

Theorem 8.3.1. Given a point on a surface, if two principal paths are both characteristic paths, then the point is a prominent feature point.

Proof. For a point p on surface S, the second fundamental form of S at p is given by the quadratic form

$$Q(\mathbf{v}) = -\mathbf{n}'(\mathbf{v}) \cdot \mathbf{v}$$

where $\mathbf{n}$ is the surface normal at p and $\mathbf{v}$ is a surface tangent. Geometrically, the linear mapping $\mathbf{n}(\mathbf{v})$ is the differential of the surface normal in the direction of $\mathbf{v}$ and $Q(\mathbf{v})$ is the normal curvature of the curve $\mathbf{c}(s)$ on S passing through $p = \mathbf{c}(0)$ with tangent $\mathbf{v} = \mathbf{c}'(0)$.

For the curve $\mathbf{c}(s)$, let $\mathbf{n}(s)$ be the surface normal on the curve $\mathbf{c}(s)$ and $Q(s)$ be the curvature on $\mathbf{c}(s)$ (i.e., $\kappa(s) = Q(\mathbf{c}'(s))$). We have

$$\kappa'(s) = \frac{dQ(s)}{ds} = -\frac{d}{ds}(\mathbf{n}'(s) \cdot \mathbf{c}'(s)) = -(\mathbf{n}''(s) \cdot \mathbf{c}'(s) + \mathbf{n}'(s) \cdot \mathbf{c}''(s)).$$

Since $\mathbf{c}''(s) = \kappa\mathbf{n}(s)$ and $\mathbf{n}'(s) = -\kappa\mathbf{c}'(s)$ (Frenet formulas), the second term vanishes identically and we arrive at

$$\kappa'(s) = -\mathbf{n}''(s) \cdot \mathbf{c}'(s). \tag{8.11}$$

Let the two principal directions at p be $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$. If the tangent $\mathbf{v} = \mathbf{c}'(0)$ forms an angle θ from $\mathbf{e}_1$ and κ_1 and κ_2 are two principal curvatures along $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$, respectively, we can derive $\kappa'(0)$ as follows:

$$\begin{aligned}\kappa'(0) &= -\mathbf{n}''(\cos\theta\hat{\mathbf{e}}_1 + \sin\theta\hat{\mathbf{e}}_2) \cdot (\cos\theta\hat{\mathbf{e}}_1 + \sin\theta\hat{\mathbf{e}}_2) \\ &= -(\mathbf{n}''(\hat{\mathbf{e}}_1) \cdot \hat{\mathbf{e}}_1)\cos^2\theta - (\mathbf{n}''(\hat{\mathbf{e}}_2) \cdot \hat{\mathbf{e}}_2)\sin^2\theta \\ &= \kappa_1'(0)\cos^2\theta + \kappa_2'(0)\sin^2\theta\end{aligned} \tag{8.12}$$

The last expression follows from Eq. (8.11).

Since the two principal directions are in the tangent directions of two characteristic paths by assumption, $\kappa_1'(0) = \kappa_2'(0) = 0$. Hence we have $\kappa'(0) = 0$ along any direction $\mathbf{v}$ at p. □

It was noted before that the principal directions and curvatures can be determined from three views of a surface point. However, when the observer moves and observes continuously, the principal directions can be determined with only two views.

Proposition 8.3.4. The principal directions of a point on a smooth surface can be determined by two views made by an observer moving continuously.

Proof. Given two principal curvatures κ_1, κ_2, Euler's formula states that any normal curvature

κ in the direction of θ to one of the principal directions is given by

$$\kappa = \kappa_1 \cos^2\theta + \kappa_2 \sin^2\theta.$$

Differentiate and we get

$$\frac{d\kappa}{d\theta} = (\kappa_2 - \kappa_1)\sin 2\theta.$$

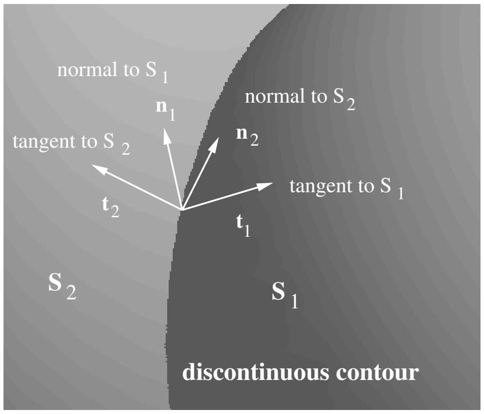
Since this is an equation of the difference of principal curvatures $(\kappa_2 - \kappa_1)$ and θ, only two views are needed to solve it. $\square$

A feature point by itself is a local concept which may not be obvious to an observer. However, the next proposition shows that it is part of two more readily identifiable features: feature curves and prominent feature points.

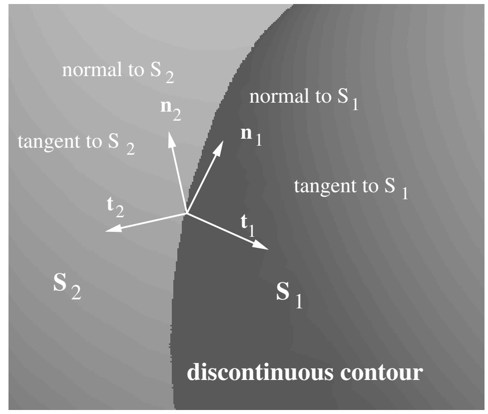
Proposition 8.3.5. A feature point on a surface is either part of a feature curve or a prominent feature point.

8.3.2.3 Visibility of Surface

A discontinuous contour can be modeled as a piecewise-planar curve in space. Given a planar segment, the tangent of the curve and surface tangents of the two surfaces that are normal to the tangent determine the visibility of the surface from any given vantage point. We have already assumed the conditions that the surface shape can be recovered, if visible. Hence the problem is to move to the part of space where the occluded part of the surface becomes visible to the observer. Since this is a discontinuous contour, there are at least two orthogonal triplet systems at a given point on the segment. The condition that a surface is occluded occurs when the quadrant encompassed by the surface tangent and surface normal does not overlap the quadrant formed by the other surface (Figure 8.22), i.e., $\mathbf{n}_1 \cdot \mathbf{t}_2 < 0$ (this implies $\mathbf{n}_2 \cdot \mathbf{t}_1 < 0$). For $i = 1, 2$, the visibility of surface S_i is constrained by the half plane defined by $\mathbf{t}_i$ and $\mathbf{n}_i$.



(a) Discontinuous contour with non-occluding geometry.



(b) Discontinuous contour with occluding geometry.

Figure 8.22: Surface geometry in the presence of a discontinuous contour.

In the following we treat individual navigations induced by apparent contours and by discontinuous contours.

8.3.3 Navigation Induced by Apparent Contours

Since the resolution of identifiable feature sets is controlled by scale space, a representation using the feature sets is global and the properties of scale space ([114]) ensure that we will not miss any features. On the other hand, since the feature points and their derived features are based on surface curves, every visible feature point should be covered by a collection of surface curves in the representation.

Covering a surface can be done theoretically by parameterizing the surface mathematically regardless of the distribution of features, as is the case in computer graphics. For image processing, this is analogous to uniform sampling according to the Nyquist rate regardless of the spatial distribution of the image intensity (cf. Section 2.2.2.2). The advantage of these schemes is the systematic recovery of the information being represented—object surfaces or, for image processing, images. However, the representation has no perceptual context and the components (e.g., knots in a spline or a layer within a multi-resolution pyramid) used in the representation have no perceptual meaning. Furthermore, the efficiency of completing a task is not taken into consideration.

An apparent contour is also a surface curve that makes the feature points on the curve explicit for observation. It also provides possible hypotheses regarding surface shape. These two aspects are essential for active navigation and will be discussed next.

8.3.3.1 Shape Recovery

In this section it is shown that a natural mesh is formed when the observer navigates the surface using movement induced by apparent contours and external references. The density of the mesh is a function of the scale and the distribution of surface features. The end result is the recovery of all the apparently visible (see Definition 8.1.1) part of the surface. In the case where the surface is appropriately textured, the entire visible surface can be recovered by navigation.

With the aid of an external reference, the observer can plan a movement so that the apparently visible part of the surface can be surveyed systematically. For each apparent contour being observed, all the feature points on the curve and their types can be identified as follows. Since every feature point on the contour can be part of a feature curve or an prominent feature point or both, goal-oriented navigation is required in order to differentiate between these alternatives. From Theorem 8.3.1, it is necessary to find two principal paths for the point. On the other hand, it is sufficient to find a non-characteristic path to disqualify the point as a prominent feature point (i.e., it is part of a feature curve). In either case, the observer gains best advantage when moving in the direction orthogonal to both the surface normal and line of sight, while keeping a constant distance from the point under examination. This is because the maximum change of view is achieved by moving orthogonal to the line of sight and this action traces out a circle around the point. Hence the observer essentially rotates around the surface. This is consistent with our experience.

Since adjacent feature points are separated by a distance determined by the scale (Proposition 8.3.1), the surface curve formed by each apparent contour will also be separated by the same distance. In other words, the surface shape within this distance is "indistinguishable." Furthermore, since the number of prominent feature points is finite (Proposition 8.3.2), the navigation process must stop deterministically.

The example in Figure 8.23 shows a prominent feature point and the recovered surface around it using the procedure just described. Figure 8.24 displays the triangular patches used as part of tracking the apparent contour.

Figure 8.23: Feature point on object surface and the feature patch.

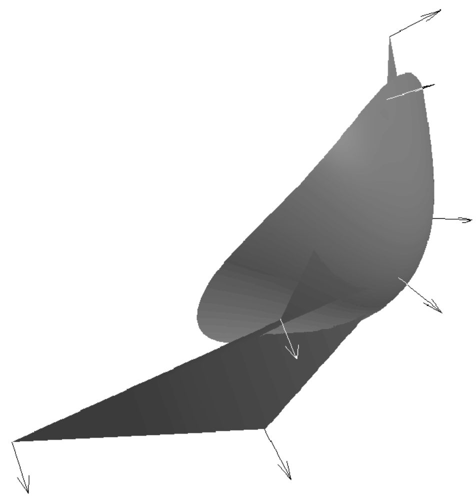


Figure 8.24: The triangular patches, surface normals, and recovered surface.

Proposition 8.3.6. The apparently visible part of a surface can be fully recovered by navigation using apparent contours.

Hence the primary navigation effort induced by observing the apparent contours is to identify prominent feature points. These features are not only useful for serving as object-centered reference frames during navigation but also for contributing to the ensemble of "parts" from which the surface is composed. However, acting alone, apparent contours are somewhat weak for surface recovery and incur a lot more effort than required with textured surfaces or when there is shading information to help form hypotheses regarding the surface shape.

8.3.3.2 Hypothesis Verification

Other than navigating to systematically acquire unknown information for potential feature points, apparent contours also motivate an observer to move in order to verify hypotheses. Since an apparent contour tells the observer partial information about the surface shape, plausible hypotheses can be formed regarding other parts of the surface. One useful hypothesis is the constancy of the apparent contour when moving in a certain direction, e.g., a translational or rotational invariance relative to a fixed observer frame. If the hypothesis is valid, this frame can be determined by the aforementioned direction of motion.

Proposition 8.3.7. An object surface is rotationally invariant if the observer acquires an invariant projection of the apparent contour during a rotational movement. The axis of rotational symmetry is the center of the rotational movement. The same is true for translational invariance and the direction of translational invariance is the same direction as the movement, which is coincident to one of the lines of sight for a fixed point on the contour.

It should be noted that for orthographic projection, the observer will not be able to discern these two kinds of invariance. For perspective projection, the computational precision of rotational invariance is proportional to the distance to the center of the symmetry.

8.3.4 Navigation Induced by Discontinuous Contours

There are two surfaces intersecting at a discontinuous contour. In addition to being a static feature, a discontinuous contour is where the shape on one side of the contour cannot be inferred from the shape on the other side.

Similar to the case of prominent feature points, discontinuous contours serve both as navigation aids and boundaries where parts of the surface meet. The shape of the component parts delimited fully or partially by discontinuous contours can be determined individually by navigation or by other means described in this chapter.

8.3.4.1 Shape Recovery

Discontinuous contours are essentially static surface marks, and share the same properties as static marks (Chapter 6), i.e., they can be used to recover surface shape around the contour when there are other static marks intersecting at points on the contour. The method of recovering the curve geometry for the contour was given in Propositions 6.2.1 and 6.2.2. In addition, occluding, discontinuous contours have one other advantage. These contours occur when the angle between the normal vector for one side and the tangent vector for the opposite side is greater than $\pi/2$, i.e., either $\mathbf{n}_1 \cdot \mathbf{t}_2 < 0$ or $\mathbf{n}_2 \cdot \mathbf{t}_1 < 0$ on a given plane (Figure 8.22). In this case, if permitted by other physical constraints, the observer can move to positions with the line of sight coinciding with the tangent of either of the surfaces. This condition is guaranteed since the reverse direction of the surface tangent falls into the half plane of visibility for the other surface.

8.3.4.2 Navigation Aid

The static nature of discontinuous contours enables the observer to navigate without external references. Furthermore, the observer knows exactly which part of the surface is being examined, and is able to shift focus whenever necessary. To use the contour for these purposes, the goal of navigation is to move to a vantage point where the surface on the other side of the contour becomes visible, if not already so, using the contour itself as a navigation landmark. In the case where surface shape can only be computed from apparent contours, navigation will have to enable the apparent contour to emerge on the other side of the discontinuous contour, i.e., the relationship between two contours will have to be observed and tracked at the same time. The general scenario will be the initial separation of two contours with the apparent one embedded completely on one side of the surface. The discontinuous one is generally invisible to the observer at this point. Hence the observer will have to know when these two contours intersect as part of the apparent contour disappears and the discontinuous one emerges. This is followed by a complete replacement of the apparent contour by the discontinuous one. Eventually the scenario reverses itself and the apparent contour emerges again at the other side of the surface.

Hence, the major motivation of navigation in the presence of discontinuous contours is to steer focus toward regions demarcated by the contour, and recover surface shape along the contour when there is occlusion. This is accomplished by moving to positions that permit full view of the surface without the obstruction of the contour while using the contour as a reference frame during the navigation.

8.4 Summary

There are two major themes in this chapter and the thesis in general: (1) how to represent the shape of an object in a global, perceptually meaningful and complete way, and (2) how to identify each individual component used in the representation, given a mobile observer.

The problem of representation for 2D curves was studied in Chapter 3, where it was shown that the extremum curvature points can be used to represent a curve comprehensively by methods such as Hermite interpolation. The same rationale was extended in this chapter to 3D surfaces by considering all possible planar curves that pass through a given surface point. Feature points and curves thus identified can be used to reconstruct the surface and the procedure is algorithmically the inverse of the representation process. The representation is also "perceptually complete" in the sense that the surface and its representation present the same set of perceptual features to the observer. In other words, the original shape information is "similar" to what can be computed from the representation when the criterion of similarity is defined by the second-order computation of surface shape. The procedure for representing the surface indicates that efficient computational procedures are available for constructing the surface from the representation.

We have established the connection between low-level visual modules and local surface shape in previous chapters. This relationship was established within the framework of scale space, which, in turn, provides a means to handle surface shape and its related operations globally. The scheme that embodies this global representation as part of the perceptual process is formulated in the first part of this chapter. In the second part of the chapter, the issue of acquiring the geometric features necessary for the representation is studied in the form of global navigation, in which the observer can systematically engage in voluntary motion guided by the current observation of various surface characteristics as projected onto the image plane. These observer motions are motivated either by the desire to identify critical features for the representation which is only partially known, or by the effort to test hypotheses formed when the observation is made. In either case, the end result is a full coverage of the object surface, and the process of exploring the surface depends on whether the surface is properly textured or if external references are available.

Chapter 9

Conclusions and Future Work

A problem in computational vision has two core parts: the task to be accomplished, and the computation needed to process the raw data in order to perform the task. When the task is related to 3D shape representation, the conventional method is to treat the problem as an inverse problem of computer graphics, in which the task is to describe surfaces using the language of computer graphics, while the required computation is to recover the depth map of the surface within an object-centered reference frame. The major problem of this approach is, except for the front-end modeling, the formulation quickly swerves away from visual perception and delves into mathematical problems such as global optimization and parametric descriptions of surface. The central thesis of this research is a direct response to this disparity of method:

- *The language employed by visual perception to represent objects is intrinsically both perceptual and geometric, and this nature has to be reflected in all stages of information processing*.
- *The nature of visual information processing is active rather than passive*.

In the light of these statements, five problems were raised as core problems in computational vision for 3D representation. This research contributes to each of them as follows.

9.1 Thesis Contributions

Local Geometric Computation

- What are the essential computations at the front-end of a vision system that is capable of computing representations for complex shapes

For tasks involving shape representation, the essential computation at the front-end is to select the geometric features that are essential for all the tasks that the system is able to accomplish. For tasks related to active navigation, optical flow is also computed using a similar mechanism. These computations are carried out using filters in the form of receptive fields, which constitute the primary spatio-temporal sampling mechanism.

Perception and Differential Geometry

- What kind of geometric language can be used to describe relevant perceptual results in human vision and how can the elements of the language be computed from the data received at the front-end?

The local computations carried out at the front-end necessarily dictate a formal formulation using a differential language. The components of the language include tangent, curvature, and derivative of curvature along 2D contours. Efficient and stable methods for computing these invariants locally and directly from raw images were derived as a result of the study. The formulations and methods for these computations are all novel.

Global Representation from Local Computation

- How the global properties of perception are related to the results of local computation as dictated by our choice of geometric language?

The differential geometric language is naturally embedded in a scale space intrinsic to the sampling topology of the front-end. All the global properties of visual perception can be derived within this scale space by using the local components of the geometric language. This is a consequence of both the choice of the receptive fields and the formal properties of the scale space.

Incremental 3D Modeling

- When should the computation of the language terminate and what is the relationship between the representations computed in different resource constraint?

The computation for a specific task terminates when available computational resources can not accommodate the computation. The resulting models of this computation relate to other models of different computational resources through incremental modeling. Within this model, any additional local features will only affect local shape of the representation. The perception-based representation of 3D surfaces that embodies incremental modeling is a contribution of the thesis that is completely new.

Active Navigation

- How can an autonomous system actively seek out information based on what has already been observed?

The information necessary to complete a given task is dependent on the nature of the task. In the case of shape representation, the information can be actively acquired through global navigation. The formal properties of scale space guarantee the termination of the navigation as a computational process. For a given scale, associated stationary and apparent contours can be used effectively to identify feature points on the surface, which is the result of both geometric computation and visual perception. Alternative shapes that are consistent with current observations can be hypothesized and verified by navigation as well. In cases when the surface is textured, optical flow and local texture tracking provide efficient means for navigation and surface recovery. To relate the representation of global surface shape to both visual perception and efficient navigation methods is a unique contribution of the thesis.

9.2 Future Work

Stability of Computation

How a given computational framework will behave in a natural environment is a vital part of the research in computational vision. This stability issue is especially critical for a differential framework. In this research, it has been demonstrated that local geometric computations of high-order differentials are stable, and this stability is inherited by the global representation using the results from local computations. However, the same study should be carried out for optical flow computation as well as 3D shape representation. In addition, the scale space formulation can also be useful in controlling the stability of the computation, which needs to be studied further.

2D Matching

The problems of stable and information-preserving representation, and efficient computation for 2D matching are studied in this research. However, there are a large number of practical applications requiring the matching of 2D shapes. A general matching mechanism is needed which that has constant-time matching complexity independent of the database size, and is invariant under various viewpoint-dependent transformations (especially scaling and rigid transformations).

Efficient Navigation

There are several propositions and theorems proved in this thesis regarding how to navigate around a surface using current knowledge of the surface. However, efficient navigation requires appropriate reference frames as well as route planning. These problems need to be studied further and real systems need to be built to explore the efficacy of the methods.

3D Matching

Since the feature points used for 3D representation are also salient for visual perception, the matching for 3D object recognition can use these feature points and their local geometric properties as inputs. This matching process is also one of the core parts in an object recognition system, which needs to be studied as a testbed for the theory presented here.

Symbols and Information Structure

Specific to humans and primates is the ability to manipulate symbols, and to interact with the environment as a result of this manipulation. Both are characteristics of intelligence. Hence, the study of the relationships between symbol manipulation and perception is essential in the general domain of artificial intelligence. From the perspective of perception, a symbol represents essentially a piece of information that is invariant to some general contexts. Consequently, it is closely related to the invariant structure of the information. Since invariance is a major part of visual representation, it is natural to study how a category in visual perception can be represented by an information structure that is invariant under certain criteria, and, henceforth, can be represented by a symbol.

Appendix A

Curvature and Its Gradient in Observer Frame

A.1 Projected Curvature in the Observer Frame

Let the curve C on object surface be parameterized by its curve length s as $(x(s), y(s), z(s))$. The projection of C onto the image plane is a 2D curve C_p parameterized by $(\xi(t), \eta(t))$. From the imaging model for projection we have the standard projective equations:

$$\begin{aligned} \xi(s) &= \frac{x(s)}{z(s)} \\ \eta(s) &= \frac{y(s)}{z(s)}. \end{aligned} \tag{A.1}$$

Since the natural parameter s of C becomes a general parameter t of C_p, the projected curvature κ_p of C_p is:

$$\kappa_p = \frac{|\xi'\eta'' - \xi''\eta'|}{[(\xi')^2 + (\eta')^2]^{3/2}}. \tag{A.2}$$

Because C is parameterized by curve length, from the definition of the Frenet frame we have:

$$\begin{aligned}\hat{\mathbf{t}} &= (x', y', z') \\ \hat{\mathbf{n}} &= \frac{1}{\kappa}(x'', y'', z'') \\ \hat{\mathbf{b}} &= \hat{\mathbf{t}} \times \hat{\mathbf{n}} = \frac{1}{\kappa}(y'z'' - z'y'', z'x'' - x'z'', x'y'' - y'x'') = (b_1, b_2, b_3),\end{aligned} \tag{A.3}$$

where κ is the curvature of C at the point P (represented by $\mathbf{r}$ in the observer frame). Let $\mathbf{c} \triangleq \mathbf{r} \times \hat{\mathbf{t}} = (c_1, c_2, c_3)$. Substituting the differentials in Eq. (A.1) into Eq. (A.2) and using Eq. (A.3), we have

$$\kappa_p = \frac{|c_1 x'' + c_2 y'' + c_3 z''|}{[(c_1^2 + c_2^2)/z^2]^{3/2}}. \tag{A.4}$$

The quantity $(c_1^2 + c_2^2)$ is the length of $\mathbf{c}_p$, where $\mathbf{c}_p$ is the projection of $\mathbf{c}$ onto the image plane and z is the component of $\mathbf{r}$ in the $\hat{\mathbf{z}}$ direction. Using the component form of the cross product $\mathbf{c} = \mathbf{r} \times \hat{\mathbf{t}}$ and Eq. (A.3) we can derive

$$c_1 x'' + c_2 y'' + c_3 z'' = \kappa \mathbf{r} \cdot \hat{\mathbf{b}}. \tag{A.5}$$

The denominator of Eq. (A.4) can be rewritten as

$$\frac{(c_1^2 + c_2^2)}{z^2} = \frac{|\mathbf{c}|^2 - (\mathbf{c} \cdot \hat{\mathbf{z}})}{(\mathbf{r} \cdot \hat{\mathbf{z}})^2} = \frac{|\mathbf{r} \times \hat{\mathbf{t}}|^2 - (\mathbf{r}, \hat{\mathbf{t}}, \hat{\mathbf{z}})^2}{(\mathbf{r} \cdot \hat{\mathbf{z}})^2}, \tag{A.6}$$

where $(\mathbf{r}, \hat{\mathbf{t}}, \hat{\mathbf{z}})$ is a shorthand for $\mathbf{r} \times \hat{\mathbf{t}} \cdot \hat{\mathbf{z}}$. Hence Eq. (A.4) takes the vector form:

$$\kappa_p = \frac{\kappa |\mathbf{r} \cdot \hat{\mathbf{b}}|}{[(|\mathbf{r} \times \hat{\mathbf{t}})|^2 - (\mathbf{r}, \hat{\mathbf{t}}, \hat{\mathbf{z}})^2)/(\mathbf{r} \cdot \hat{\mathbf{z}})^2]^{3/2}}. \tag{A.7}$$

A.2 Projected Curvature Gradient in the Object Frame

The scalar field $\kappa_p(\mathbf{r}^*)$ given by Eq. (6.2) has steepest rate of change along the direction of its gradient, $\nabla\kappa_p$, and the change in an arbitrary direction $\mathbf{r}$ is given by $\nabla\kappa_p \cdot \mathbf{r}$. Let

$$A = \frac{|\mathbf{r}^* \times \hat{\mathbf{t}}|^2 - (\mathbf{r}^*, \hat{\mathbf{t}}, \hat{\mathbf{z}})^2}{(\mathbf{r}^* \cdot \hat{\mathbf{z}})^2}.$$

Then the gradient of κ_p in the object frame $\nabla\kappa_p(\mathbf{r}^*)$ takes the form

$$\nabla\kappa_p(\mathbf{r}^*) = \pm\frac{\kappa}{A^{5/2}}\left[A\hat{\mathbf{b}} + 3(\mathbf{r}^* \cdot \hat{\mathbf{b}})\frac{z'}{z^{*2}}(c_2^*, -c_1^*, -\frac{c_2^* x^* - c_1^* y^*}{z^*})\right]. \tag{A.8}$$

Note that the sign of the expression depends on the sign of $(\mathbf{r}^* \cdot \hat{\mathbf{b}})$ and that z' is the third component of $\hat{\mathbf{t}}$, which is identical in both frames. Define

$$\mathbf{r}_\perp^* \triangleq (c_2^*, -c_1^*, -\frac{c_2^* x^* - c_1^* y^*}{z^*}). \tag{A.9}$$

which, by its form, denotes a vector orthogonal to $\mathbf{r}^*$ since $\mathbf{r}^* \cdot \mathbf{r}_\perp^* = 0$. Eq. (A.8) can then be expressed as

$$\nabla\kappa_p(\mathbf{r}^*) = \pm\frac{\kappa}{A^{5/2}}\left[A\hat{\mathbf{b}} + 3(\mathbf{r}^* \cdot \hat{\mathbf{b}})\frac{z'}{z^{*2}}\mathbf{r}_\perp^*\right]. \tag{A.10}$$

Since $\mathbf{r}^* = -\mathbf{r}$ and $\mathbf{r}_\perp^* = -\mathbf{r}_\perp$ we have $\nabla\kappa_p(\mathbf{r}) = \nabla\kappa_p(\mathbf{r}^*)$. This expresses the fact that the relative translational motion of object and observer is indistinguishable. But this does not carry over to rotational motion (see next section). Now let's consider the way κ_p changes along $\mathbf{c}_p^*$, i.e., consider the expression $\nabla\kappa_p \cdot \mathbf{c}_p^*$. Since $\mathbf{c}_p^* \cdot \mathbf{r}_\perp^* = 0$ we have

$$\nabla\kappa_p \cdot \mathbf{c}_p^* = \pm\frac{\kappa}{A^{3/2}}\mathbf{c}_p^* \cdot \hat{\mathbf{b}}.$$

Furthermore, using Eq. (6.1) and $\mathbf{c}_p^* = -\mathbf{c}_p$, $\mathbf{r}^* = -\mathbf{r}$, we have

$$\nabla\kappa_p \cdot \mathbf{c}_p^* = -\nabla\kappa_p \cdot \mathbf{c}_p = \kappa_p \frac{\mathbf{c}_p^* \cdot \hat{\mathbf{b}}}{\mathbf{r}^* \cdot \hat{\mathbf{b}}} = \kappa_p \frac{\mathbf{c}_p \cdot \hat{\mathbf{b}}}{\mathbf{r} \cdot \hat{\mathbf{b}}}. \tag{A.11}$$

A.3 Proof of Proposition 6.2.1

Proof. The space outside a convex surface is defined by $\mathbf{r} \cdot \hat{\mathbf{n}} > 0$. Consider region I where $\mathbf{r} \cdot \hat{\mathbf{b}} < 0$. Since $\mathbf{c} = \mathbf{r} \times \hat{\mathbf{t}}$ and $\hat{\mathbf{t}} = \hat{\mathbf{n}} \times \hat{\mathbf{b}}$, we have

$$\mathbf{c} = (\mathbf{r} \cdot \hat{\mathbf{b}})\hat{\mathbf{n}} - (\mathbf{r} \cdot \hat{\mathbf{n}})\hat{\mathbf{b}}. \tag{A.12}$$

The vector $\mathbf{c}_p$ is the projection of $\mathbf{c}$ onto the image plane and they are related through $\mathbf{c} = \mathbf{c}_p + (\mathbf{c} \cdot \hat{\mathbf{z}})\hat{\mathbf{z}}$. Applying Eq. (A.12), we have

$$\mathbf{c} = \mathbf{c}_p + \Big[(\mathbf{r} \cdot \hat{\mathbf{b}})(\hat{\mathbf{n}} \cdot \hat{\mathbf{z}}) - (\mathbf{r} \cdot \hat{\mathbf{n}})(\hat{\mathbf{b}} \cdot \hat{\mathbf{z}})\Big]\hat{\mathbf{z}}. \tag{A.13}$$

Hence

$$\begin{aligned}
\mathbf{c}_p \cdot \hat{\mathbf{b}} &= \mathbf{c} \cdot \hat{\mathbf{b}} - \Big[(\mathbf{r} \cdot \hat{\mathbf{b}})(\hat{\mathbf{z}} \cdot \hat{\mathbf{n}}) - (\mathbf{r} \cdot \hat{\mathbf{n}})(\hat{\mathbf{z}} \cdot \hat{\mathbf{b}})\Big](\hat{\mathbf{z}} \cdot \hat{\mathbf{b}}) \\
&= -(\mathbf{r} \cdot \hat{\mathbf{n}}) - (\mathbf{r} \cdot \hat{\mathbf{b}})(\hat{\mathbf{z}} \cdot \hat{\mathbf{n}})(\hat{\mathbf{z}} \cdot \hat{\mathbf{b}}) + (\mathbf{r} \cdot \hat{\mathbf{n}})(\hat{\mathbf{z}} \cdot \hat{\mathbf{b}})^2 \\
&= (\mathbf{r} \cdot \hat{\mathbf{n}})\Big[(\hat{\mathbf{z}} \cdot \hat{\mathbf{b}})^2 - 1\Big] - (\mathbf{r} \cdot \hat{\mathbf{b}})(\hat{\mathbf{z}} \cdot \hat{\mathbf{n}})(\hat{\mathbf{z}} \cdot \hat{\mathbf{b}}) \\
&\triangleq \alpha - \beta.
\end{aligned}$$

From Cauchy's inequality, we have

$$\hat{\mathbf{z}} \cdot \hat{\mathbf{b}} \le |\hat{\mathbf{z}}||\hat{\mathbf{b}}| = 1.$$

Since the observer is in region I and observes a convex surface from an agreeable frame, we have $\mathbf{r} \cdot \hat{\mathbf{n}} > 0$ (convex), $\mathbf{r} \cdot \hat{\mathbf{b}} < 0$ (region I), and $\hat{\mathbf{z}} \cdot \hat{\mathbf{n}} > 0, \hat{\mathbf{z}} \cdot \hat{\mathbf{b}} < 0$ (agreeable frame). It follows that $\alpha < 0$ (the first part of the equation) and $\beta > 0$ (the second part of the equation). Hence $\mathbf{c}_p \cdot \hat{\mathbf{b}} < 0$. Consequently, for the observer in the agreeable observer frame, the change in κ_p in the direction of $\mathbf{c}_p$ is

$$\nabla \kappa_p \cdot \mathbf{c}_p = -\kappa_p \frac{\mathbf{c}_p \cdot \hat{\mathbf{b}}}{\mathbf{r} \cdot \hat{\mathbf{b}}},$$

which is always negative. Similar arguments hold for region II and for concave surfaces. $\square$

Bibliography

[1] E. H. Adelson and J. R. Bergen. Spatialtemporal energy models for the perception of motion. *J. Opt. Soc. Amer. A*, 2(2):284–299, 1985.

[2] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. Patt. Anal. Machine Intell.*, 1985.

[3] J. Allmen and E. McGuinness F. Miezin. Simulus specific responses from beyond the classical receptive field: Neurophysiological mechanisms for local-global comparisons in vision neurons. *Annual Reviews of Neuroscience*, 8:407–430, 1985.

[4] J. Aloimonos and D. Shulman. *Integration of Visual Modules*. Academisc Press, 1989. An extension of Marr's paradigm.

[5] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. In *Proc. of 1st Int. Conf. on Computer Vision*, pages 35–54, 1987.

[6] K. Arbter. Affine-invariant Fourier descriptors. In J. C. Simon, editor, *From Pixels to Features*, pages 153–164. Elsevier Science, 1989.

[7] H. Asada and M. Brady. The curvature primal sketch. AI Memo 758, MIT, 1984.

[8] F. Attneave. Some informational aspects of visual perception. *Psychology Review*, 61:183–193, 1954.

[9] A. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1:1–20, 1981.

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

[11] H. Barrow and J. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17:75–116, 1981.

[12] J. S. Beis and D. G. Lowe. Learning indexing functions for 3-D model-based object recognition. In *Proc. CVPR94*, pages 275–280, 1994.

[13] M. Bertero, T. Poggio, and V. Torre. Ill-posed problems in early vision. AI Memo 924, MIT, May 1987.

[14] P. Besl and R. Jain. Invariant surface characteristics for 3d object recognition in range images. *Comput. Vis. Graph. and Image Process.*, 33:33–80, 1986.

[15] G. Birkhoff and G. Rota. *Ordinary Differential Equations*. Wiley, 1978.

[16] A. Black. Reconstructing a visible surface. In *Proc. Nat. Conf. Artificial Intell.*, pages 23–26, 1984.

[17] A. Blake, A. Zisserman, and R. Cipolla. Visual exploration of free-space. In A. Blake and A. Yuille, editors, *Active Vision*. MIT Press, 1992.

[18] R. M. Bolle and B. C. Vemuri. On three-dimensional surface reconstruction methods. *IEEE Trans. Patt. Anal. Machine Intell.*, 13(1):1–13, 1991.

[19] M. Brady, J. Ponce, A. Yuille, and H. Asada. Describing surfaces. *Computer Vision, Graphics, and Image Processing*, 32:1–28, 1985.

[20] B. G. Breitmeyer. Eye movements and visual pattern perception. In *Pattern Recognition for Human and Machine: Visual Perception*. Academic Press, 1986.

[21] A. R. Bruss and B. K. P. Horn. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 21, 1983.

[22] J. Canny. A computational approach to edge detection. *IEEE Trans. Patt. Anal. Machine Intell.*, 8(6):679–698, 1986.

[23] R. Cipolla. *Active Visual Inference of Surface Shape*. Springer-Verlag, Heidelberg, Germany, 1996.

[24] R. Cipolla and A. Blake. Motion planning using image divergence and deformation. In A. Blake and A. Yuille, editors, *Active Vision*. MIT Press, 1992.

[25] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *Int. J. Computer Vision*, 9(2):83–112, 1992.

[26] R. Cipolla and A. Zisserman. Qualitative surface shape from deformation of image curves. *Int. J. Computer Vision*, 8(1):53–69, 1992.

[27] J. G. Daugman. Two dimensional spectral analysis of cortical receptive field profile. *Vision Research*, 20:847–856, 1980.

[28] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency. and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Amer.*, 2(7):1160–1169, 1985.

[29] R. L. DeValois, D. G. Albrecht, and L. G. Thorel. Spatial frequency selectivity of cells in macaque visual cortex. *Vision Research*, 22:545–559, 1982.

[30] R. L. DeValois, E. W. Yund, and N. Hepler. The orientation and direction selectivity of cells in macaque visual cortex. *Vision Research*, 22:531–544, 1982.

[31] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.

[32] A. Dobbins, S. Zucker, and M. Cynader. Endstopping and curvature. *Vision Research*, 29(10):1371–1387, 1989.

[33] G. Farin. *Curves and Surfaces For Computer Aided Geometric Design*. Academic Press, Third edition, 1993.

[34] O. Faugeras, M Herbert, and E. Pauchon. Segmentation of planar and quadratic patches from range data. In *Proc. IEEE Conf. Pattern Recognition and Image Processing*, 1983.

[35] M. Fischler and R. Bolles. Perceptual organization and curve partitioning. *IEEE Trans. Patt. Anal. Machine Intell.*, 8(1):100–105, 1986.

[36] M. Fischler and H. Wolf. Locating perceptually salient points on planar curves. *IEEE Trans. Patt. Anal. Machine Intell.*, 16(2):113–129, 1994.

[37] L. M. Florack, B. M. H. Romeny, J. Koenderink, and M. A. Viergever. Scale and the differential structure of images. *Image and Vision Computing*, 1992.

[38] D. Gabor. Theory of communication. *Journal Inst. Electrical Engineering*, 93:429–457, 1946.

[39] P. Giblin and R. Weiss. Reconstruction of surfaces from profiles. In *Proc. 1st Int. Conf. on Computer Vision*, pages 136–144, 1987.

[40] H. Goldstein. *Classical Mechanics*. Addison-Wesley, 2nd edition, 1980.

[41] W. E. L. Grimson. On the recognition of parametrized 2D objects. *Int. J. Computer Vision*, 3:353–372, 1989.

[42] S. Grossberg and M. E. Rudd. A neural architecture for visual motion perception: Group and element apparent motion. *Neural Networks*, 2:421–450, 1989.

[43] N. M. Grzywacz and T. Poggio. Computation of motion by real neurons. In *An Introduction to Neural and Electronic Networks*. Academic Press, 1990.

[44] B. Hamann. Curvature approximation for triangulated surfaces. In G. Farin, H. Hagen, and H. Noltemeier, editors, *Geometric Modeling*. Springer-Verlag Wien New York, 1993.

[45] D. Heeger. Model for the extraction of image flow. *J. Opt. Soc. Amer. A*, 4(8), 1987.

[46] D. J. Heeger. Visual perception of three-dimensional motion. *Neural Computation*, 2:129–137, 1990.

[47] E. C. Hildreth and C. Koch. The analysis of visual motion: From computational theory to neuronal mechanisms. AI Memo 919, MIT, December 1986.

[48] D. Hoffman and W. Richards. Parts of recognition. *Cognition*, 18:65–96, 1985.

[49] D. D. Hoffman and W. A. Richards. Representing smooth plane curves for recognition: Implications for figure-ground reversal. In Whitman Richards, editor, *Natural Computation*. MIT Press, 1988.

[50] B. K. P. Horn. Extended Gaussian images. *Proc. IEEE*, 72:1671–1686, 1984.

[51] B. K. P. Horn. *Robot Vision*. MIT Press, 1986.

[52] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *J. Physiol. London*, 160:106–154, 1962.

[53] D. H. Hubel and T. N. Wiesel. Sequence regularity and geometry of orientation columns in the monkey striate cortex. *Journal of Comparative Neurology*, 158(3):267–294, 1974.

[54] R. A. Hummel. Representations based on zero-crossings in scale-space. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 204–209, 1986.

[55] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. Patt. Anal. Machine Intell.*, 15(9):850–863, 1993.

[56] Hoschek J and D. Lasser. *Computer Aided Geometric Design*. A K Peters, 1993.

[57] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, pages 259–268, 1987.

[58] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[59] J. J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984.

[60] J. J. Koenderink. *Solid Shape*. MIT Press, Cambridge, MA, 1990.

[61] J. J. Koenderink and W. Richards. Two-dimensional curvature operators. *J. Opt. Soc. Amer. A*, 5(7), 1988.

[62] J. J. Koenderink and A. J. van Doorn. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica Acta*, 22(9):773–791, 1975.

[63] J. J. Koenderink and A. J. van Doorn. The singularities of the visual mapping. *Biological Cybernetics*, 24:51–59, 1976.

[64] J. J. Koenderink and A. J. van Doorn. Optic flow. *Vision Research*, 26(1):161–180, 1986.

[65] J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[66] J. J. Koenderink and A. J. van Doorn. Receptive field families. *Biological Cybernetics*, 63:291–297, 1990.

[67] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8), 1992.

[68] M. Kunt, A. Ikonomopoulis, and M. Kocher. Second-generation image-coding techniques. *Proc. IEEE*, 73(4):549–574, 1985.

[69] K. N. Kutulakos and C. R. Dyer. Occluding contour detection using affine invariants and purposive viewpoint control. In *Computer Vision and Pattern Recognition*, pages 323–330, 1994.

[70] E. Lee. Choosing nodes in parametric curve interpolation. *Computer Aided Design*, 21(6), 1989.

[71] W. R. Levick. Sampling of information space by retinal ganglion cells. In J. D. Pettigrew and K. J. Sanderson, editors, *Visual Neuroscience*. Cambridge Univ. Press, 1986.

[72] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *Int. J. Computer Vision*, 11(3):283–318, 1993.

[73] M. Livingstone and D. Hubel. Segregation of form. color, movement, and depth: Anatomy, physiology, and perception. *Science*, pages 740–749, 1988.

[74] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proc. Roy. Soc. London B*, 208:385–397, 1980.

[75] D. M. MacKay. Vision — the capture of optical covariation. In J. D. Pettigrew and K. J. Sanderson, editors, *Visual Neuroscience*. Cambridge Univ. Press, 1986.

[76] H. A. Mallot, W. von Seelen, and F. Giannakopoulos. Neural mapping and space-variant image processing. *Neural Networks*, 3:245–263, 1990.

[77] D. Marr. *Vision*. W. H. Freeman, 1982.

[78] D. Marr, T. Poggio, and E. Hildreth. Smallest channel in early human vision. *J. Opt. Soc. Amer.*, 70(7):868–870, 1980.

[79] S. Marăelja. Mathematical description of the response of simple cortical cells. *J. Opt. Soc. Amer.*, 70(11):1297–1300, 1980.

[80] E. Milios. Shape matching using curvature processes. *Computer Vision, Graphics, and Image Processing*, 47:203–226, 1989.

[81] F. Mokhtarian and A. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Patt. Anal. Machine Intell.*, 14(8):789–805, 1992.

[82] J. L Mundy and A. Zisserman. *Geometric Invariance in Compter Vision*. MIT Press, 1992.

[83] A. Oppenheim and R. Schafer. *Digital Signal Processing*. Prentice-Hall, 1975.

[84] P. Parent and S. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Trans. Patt. Anal. Machine Intell.*, 11(8):823–839, August 1989.

[85] A. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[86] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[87] W. Richards, B. Dawson, and D. Whittington. Encoding contour shape by curvature extrema. *JOSAA*, 2:1483–1491, 1986.

[88] W. Richards, J. J. Koenderink, and D. D. Hoffman. Inferring 3d shapes from 2d silhouettes. In W. Richards, editor, *Natural Computation*. MIT Press, 1988.

[89] C. A. Rothwell. *Object Recognition through Invariant Indexing*. Oxford University Press, 1995.

[90] P. T. Sander and S. W. Zucker. Inferring surface trace and differential structure from 3-d images. *IEEE Trans. Patt. Anal. Machine Intell.*, 12(9):833–854, 1990.

[91] T. D. Sanger. Analysis of the two-dimensional receptive fields learned by the generalized hebbian algorithm in response to random input. *Biological Cybernetics*, 63:221–228, 1990.

[92] R. Shapley and P. Lennie. Spatial frequency analysis in the visual system. *Annual Review of Neuroscience*, 8:547–583, 1985.

[93] S. S. Sinha and P. J. Besl. Principle patches–A viewpoint-invariant surface description. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 7–11, 1990.

[94] H. Stark. Sampling theorems in polar coordinates. *J. Opt. Soc. Amer.*, 69(11):1519–1525, 1979.

[95] F. Stein and G. Medioni. Structural indexing: Efficient 3-D object recognition. *IEEE Trans. Patt. Anal. Machine Intell.*, 14(2), 1992.

[96] K A. Stevens. The visual interpretation of surface contours. *Artifical Intelligence*, 17:47–73, 1981.

[97] M. Subbarao. *Interpretation of Visual Motion: A Computational Study*. Morgan Kaufmann, 1988.

[98] G. Taubin and D. B. Cooper. Object recognition based on moment (of algebraic) invariants. In J. L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*. MIT Press, 1992.

[99] D. Terzopoulos. The computation of visible-surface representations. *IEEE Trans. Patt. Anal. Machine Intell.*, 10(4):417–438, 1988.

[100] C. Tomasi. *Shape and Motion from Image Streams: a Factorization Method*. PhD thesis, Carnegie Mellon University, 1991.

[101] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, 1979.

[102] R. Vaillant and O. Faugeras. Using extremal boundaries for 3-d object modeling. *IEEE Trans. Patt. Anal. Machine Intell.*, 14(2):157–173, 1992.

[103] A. S. Wallack and J. F. Canny. Efficient indexing techniques for model based sensing. In *Proc. CVPR94*, pages 259–266, 1994.

[104] H. Wässle. Sampling the visual space by retinal ganglion cells. In J. D. Pettigrew and K. J. Sanderson, editors, *Visual Neuroscience*. Cambridge Univ. Press, 1986.

[105] B. A. Watson. Detection and recognition of simple spatial forms. In O. J. Braddick and A. C. Sleigh, editors, *Physical and Biological Processing of Images*. Springer-Verlag, 1983.

[106] B. A. Watson. The cortex transform: Rapid computation of simulated neural images. *Computer Vision, Graphs, and Image Processing*, 39:311–327, 1987.

[107] B. A. Watson. Efficiency of a model human code. *J. Opt. Soc. Amer.*, 4(12):2401–2417, 1987.

[108] B. A. Watson and Jr. A. J. Ahumada. Model of human visual-motion sensing. *J. Opt. Soc. Amer. A*, 2(2):322–342, 1985.

[109] A. M. Waxman and S. Ullman. Surface structure and three-dimensional motion from image flow kinematics. *Int. J. Robotics Research*, 4(3):72–94, 1985.

[110] J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. *Int. J. Computer Vision*, 1995.

[111] H. Wechsler and L. Zimmerman. 2D invariant object recognition using distributed associative memories. *IEEE Trans. Patt. Anal. Machine Intell.*, 10(6):811–821, 1988.

[112] I. Weiss. High-order differentiation filters that work. *PAMI*, 1994.

[113] H. R. Wilson and J. R. Bergen. A four mechanism model for threshold spatial vision. *Vision Research*, 19(19–32), 1979.

[114] A. Witkin. Scale-space filtering. In *Proc. 9th Int. Joint Conf. on Artificial Intelligence*, pages 1019–1022, 1983.

[115] A. Witkin and M. Tenenbaum. On perceptual organization. In A. Pentland, editor, *From Pixel To Predicates*, chapter 7. Ablex Publishing, 1986.

[116] A. Yuille and T. Poggio. Fingerprints theorems for zero-crossings. In *Proc. of AAAI*, 1984.

[117] A. Zisserman, A. Black, C. Rothwell, L. Van Gool, and M. Van Diest. Eliciting qualitative structure from image curve deformations. In *Proc. 4th Int. Conf. on Computer Vision*, pages 340–345, 1993.

[118] S. W. Zucker. Early orientation selection: Tangent fields and the dimensionality of their support. *Computer Vision, Graphics, and Image Processing*, 32(1):74–103, 1985.

Index