1.0 Introduction

It is hard to precisely define what the "vision problem" is since such definitions in the literature have not always been consistent. A reasonable definition might be : Given an image, construct a reasonable hypothesis for the 3D scene that generated the image. Even people rarely know with absolute certainty what is actually out there, but we are able to hypothesize intelligently, given an image, about what is *probably* in the scene. It is the purpose of this paper to propose both a general heuristic and an algorithm based on the heuristic that allows for such hypothesis formation to be generated for objects in an image that have symmetric regions and can be locally approximated by a cylindrical surface. Many, if not most, man-made objects satisfy this property. The method presented in this paper is different than previous approaches to the problem in that it is both heuristic-driven and context-sensitive and is capable of recovering 3D shape modulo scale (see section 2). The algorithm relies on a context-sensitive hierarchy, developed in the paper, where certain recoverable parameters are more context-sensitive than others. Though the notion of context-sensitivity is pervasive in such fields as computational linguistics, it is often ignored or only implicitly acknowledged in the computational vision community (with the exception of perceptual organization researchers). The general rule in resolving context-sensitive parameters is that the more context-sensitive a parameter is, the later in the algorithm it should be resolved. Since the more context-sensitive a problem is, the more global the computation required to solve it, it is natural to try and solve the more local problems before attempting to solve the more global ones.

Many objects in both the man-made world and nature exhibit some degree of symmetry. Formally, an object exhibits symmetry whenever it can be divided into two or more parts which can be permuted by the application of certain isometries which leave the original object unchanged [yale88]. Two symmetry operations for planar figures are reflection and rotation. The letter M is a reflexive symmetry while the letter S is a rotational symmetry. The letter H is both reflexively (i.e., laterally) and rotationally symmetric. If a planar figure is arbitrarily rotated in three dimensions and projected orthographically onto the image plane, the projected figure is generally no longer reflexive symmetric. A projected symmetric figure can also be generated by skewing and rotating some (other) symmetric figure in the image plane (see [freid86]). Orthographic projection is assumed throughout this paper, although it should be noted that 3D shape is recovered in this paper for objects whose images were clearly taken under perspective projection. Under orthographic projection, a right angle can project onto a right, acute, or obtuse angle in the image plane. Consequently, the reflexive symmetric contour in figure 1a projects onto the *skew symmetric* contour shown in figure 1b, where *skew symmetry* is defined in [kan81] as a symmetry not necessarily perpendicular to the axis, but at a fixed angle to it. A skewed symmetry defines two directions, a skewed-symmetry axis and a skewed-transverse axis, as shown in figure 1b.

In the real world, skew symmetric contours occur often. Figure 2a is an edge image of a PUMA robot arm from the Columbia Vision and Robotics Lab, while figure 2b is a skew symmetric contour that was hand-segmented from the same image. Also shown in Figure 2b, overlaid on the skew symmetric contour, are the axes of skew recovered by SYMAN. Similarly, figure 3a shows a SUN4 SPARC workstation in the lab, figure 3b shows a hand-segmented skew symmetric contour from the same image, and figure 3c shows the *deskewed* contour with the recovered symmetry axes (again, using SYMAN). Symmetry and skew symmetry are discussed in greater detail in section 3.

This paper presents a general algorithm for recovering 3D shape (modulo scale) from an assumption of underlying 3D object symmetry. In other words, the recovery algorithm is guided by a general heuristic referred to as the *shape from objects of symmetry* heuristic (hereafter referred to as the SOS heuristic). The SOS heuristic used in the sequel can be stated as follows:

> "The projection of an object in the image is interpreted wherever possible as a symmetric object in the scene."

Although the terms symmetry and skew symmetry are defined formally in this paper (section 3), the term "symmetric object" is not. The term is difficult to define because often, in the real world, objects are not exactly symmetric but only approximately so. This motivates the need for a more general class of symmetry that takes into consideration approximate symmetry. A formal definition of this type of symmetry, termed ε-symmetry, is given in [gros90b]. The term "symmetric object" is best clarified by example; this is done in section 5. In general terms what is meant is that an interpretation of the image is given such that the surfaces or objects hypothesized to be in the scene exhibit some degree of regularity.

If the underlying shapes in the scene is presumed to have some degree of real symmetry then we can expect its image to exhibit some degree of skew symmetry. In a related paper [gros90a], methods of detecting skew symmetric contours are presented. This paper assumes that, starting with an intensity image, the edge map has been generated and that skew symmetric edges have been detected. The problem we wish to solve for then is the following : given an edge image with detected skewed contours, construct a context-sensitive algorithm to recover the underlying 3D shape assuming the object exhibits some degree of symmetry. This paper does not consider other more general definitions of symmetry that have been defined in the context of ribbons (see [rosen86]) and generalized cylinders (see [rao88]).

A method for finding axes of skewed symmetry given a skewed planar contour is an important step in the larger shape recovery algorithm. To find such axes of skewed symmetry, the algorithm presented in this paper relies on a symmetry analyzer named SYMAN [gros90a]. A review of techniques for finding axes of skew is given in [gros90b] (also see [ponce90][marc90],[yuen89][freid86]). SYMAN finds axes of skew given a skewed contour and can conjecture about certain properties of the skew contour such as the degree of skew symmetry (see [gros90a]). The SYMAN system is discussed in section 4.

Many vision researchers have focused on recovering shape from contour (i.e., edges), while ignoring other image cues, such as image intensity, for two primary reasons. The first is that humans seems very adept at shape from contour, without additional cues. The second reason is that the problem is not nearly as *context-sensitive* as the more general vision problem. Methods using image intensities to recover shape almost invariably are required to make some strong assumptions about either the imaging model (e.g., number of light sources, their positions, properties of the material surface, etc.) or the object model. The intensity image, then, is a function of many parameters and it is difficult to recover the underlying shape without knowing some of these parameters *apriori*. The shape from contour problem is less context-sensitive than the general shape from image intensity problem. While the image contour generated from a particular scene is a function of viewing angle with respect to the object-centered coordinate system, it is generally invariant (assuming some ambient light) to such imaging model parameters as number of light sources or their positions. Because the shape from contour problem has fewer parameters yet seems to be solvable, i.e., humans do it, there has been considerable focus in the vision community on this problem.

This paper describes a method that is context-sensitive, not model-based, and can theoretically be used to recover the 3D shape of an object from a single edge image where the object surface is locally cylindrical (to an approximation) and contains symmetric surface regions. Previous work in the area of shape from edges is presented in section 2. Some basic definitions of symmetry and skew symmetry are given in section 3. A description of SYMAN, the skew symmetry analyzer, is given in section 4. Given groupings (alternatively collations) of skew symmetric contours, the first priority is to find local *2D coordinate systems* for these contours; this requires solving for axes of skew. In general, the degree of skew symmetry for a skewed contour is greater than one, i.e., there is axial ambiguity, and several pairs of skew axes are recovered by SYMAN. Section 5 presents a context-sensitive method of filtering the skew solutions to recover a unique set of axes for each skew contour. Having recovered these local 2D coordinate systems, adjacent contours are then grouped into *surface collations* based on the degree of parallelism among their local coordinate systems and, using a constraint from Stevens [stev81], gradient constraints can be computed for each locally symmetric region (section 6). In section 7, a method is presented for finding an orthogonal basis for object space. Having recovered the transformation between viewer space and object space, gradient values of symmetric regions (constrained in section 6) are solved for uniquely. Conclusions and future work are described in section 8.

2.0 Previous Work

The problem of recovering shape from monocular edge images has been studied by numerous vision researchers. There have, in general, been three basic *approaches* to the problem. In this section, each of these approaches is reviewed and critiqued.

One approach to recovering shape from edges relies on heuristics. Kanade [kan81] recovers the shape of skewed contours by interpreting them as projections of real symmetries in the scene and then resolves the gradient ambiguity by selecting an interpretation that minimizes slant. Brady and Yuille [brad83] interpret a contour as a scene curve of maximum compactness. Such heuristic-based approaches fail, by definition, when the heuristics are violated. Since the actual scene parameters are generally not known *apriori*, the only way to determine if a heuristic is being violated is from additional image context. The maximum compactness heuristic interprets an ellipse in the image as a circle in the scene. But whether or not the image ellipse is a circle in the scene is generally context-sensitive and is a function of whether or not the adjacent contours are bilaterally symmetric. The elliptical contour in Figure 4 is the projected cross-section curve of a generalized cylinder and is clearly not a circular curve in the scene since the extremal edges (i.e., limbs) do not preserve bilateral symmetry (see [nal87a]).

Another class of methods aimed at recovering shape from edges can be termed model-based. A good example of this type of approach is that of ACRONYM [brook81]. In ACRONYM, the domain was restricted to a few, specific classes of generalized cylinders (which were further constrained by specific model information). Another example of model-based shape from contour recovery is that of [ponce89] where torii are recovered from edge contours. These model-based methods generally take advantage of information available from extremal and/or discontinuous edges. Such shape recovery, however, is restricted to particular classes of shape models. There are also hybrid approaches that use heuristics coupled with models, such as [hor87] and [ulup88]. In [hor87], for example, a contour is interpreted as being the projection of a maximally compact 3D curve that is consistent with being the orthogonal cross-section curve of a generalized cylinder. Such hybrid methods are still restricted to recovering shape for projections from a limited class of shape models.

A third approach is that of perceptual organization, as in lowe [lowe85] and mohan [moh89]. Perceptual organization is defined in [lowe85] as the basic capablity of the human visual system to derive relevant groupings and structures from an image without prior knowledge of its contents. Techniques in computer vision that try and mimmick these human capabilities are referred to in this paper as perceptual organization methods. Perceptual organization methods, in their attempt at simulating not just what the human visual system does but also how it does it, are quite general in scope. Specifically, by using hierarchical sets of features and comparing features across the entire image, the perceptual organization techniques used in [lowe85][moh89] are sensitive to context while at the same time are not constrained to a set of specific models (although there are some prior expectations of regularity, parallelism, etc.). Perceptual organization techniques have not actually been used to recover 3D shape, but rather to recover meaningful 2D features. In [lowe85] an organized 2D representation of the image was generated and then, to recover the underlying shape, the recovered features were matched to a set of models (e.g., stapler). In [moh89], the end result is a set of collated features (e.g., ribbons) that are useful as high-level features for stereo matching.

The approach taken in this paper most closely resembles the perceptual organization work. A general heuristic is used that is context-sensitive in that it considers how neighboring contours impact on each other. No specific shape models are assumed. A filtering technique that is similar to the method described in [lowe85] is used to resolve axial ambiguity (section 5) and to find an object basis (section 7). The method described in this paper is different from the methods described in [lowe85] and [moh89] in that 3D shape (modulo scale) is actually recovered without models or a database. In addition, a context-sensitive hierarchy is developed so that certain computations such as resolution of axial ambiguity are shown to be more local (less sensitive to context) than other computations, such as resolutiomn of gradient ambiguity (see section 6). In [lowe85] and [moh89], by contrast, the computations are generally global, taken over the entire image (e.g., constraint satisfaction networks), and consequently tend to be very computationally expensive.

3.0 Symmetry and Skew Symmetry

Since the recovery heuristic assumes underlying symmetry in the scene, some definitions of symmetry are appropriate. To understand how axes of skew are recovered by SYMAN (section 4), it is also helpful to understand the relationship between 3D rotation followed by orthographic projection and image skew transformation. Section 3.1 defines several classes of symmetry that are used in this paper. In section 3.2 the relationship between scene rotation followed orthographic projection and image skew transformation is discussed and then a definition of skew symmetry is given. Section 3.3 contains some propositions helpful in understanding the algorithm for SYMAN (section 4.1).

3.1 Symmetry: Some Definitions

Let $\vec{C}(t) \in R^2$ be a contour parameterized by t, $t \in [0,1]$. Let $\vec{C}(t) = (x(t), y(t)) = (\rho(t)\cos\theta(t), \rho(t)\sin\theta(t))$, where $\vec{C}(t)$ need not be continuous.
Let $S = \{ (\rho(t), \theta(t)) \mid (\rho(t)\cos\theta(t), \rho(t)\sin\theta(t)) \in \vec{C}(t) \}$. Assuming S is not a singleton, the contour $\vec{C}$ is symmetric with respect to the y axis exactly when

$$(\rho, \theta) \in S <=> (\rho, \pi - \theta) \in S$$

$$\equiv (x, y) \in S <=> (-x, y) \in S \ .$$

The contour $\vec{C}$ is symmetric with respect to the x axis exactly when

$$(\rho, \theta) \in S <=> (\rho, -\theta) \in S$$

$$\equiv \quad (x, y) \in S <=> (x, -y) \in S \ .$$

In general, $\vec{C}$ is defined as reflexive symmetric with respect to unit axis $\vec{A_\gamma}$, where γ is the angle between $\vec{A_\gamma}$ and the x axis, $2\pi \geq \gamma \geq 0$, *iff*

$$(\rho, \theta) \in S <=> (\rho, 2\gamma - \theta) \in S \ .$$

Two additional types of symmetry are now defined that will be useful in the sequel:

If $\vec{C}$ is reflexive symmetric with respect to unit axes $\vec{A_\gamma}$ and $\vec{A_\delta}$ and $\vec{A_\gamma} \cdot \vec{A_\delta} = 0$, then $\vec{C}$ is *doubly reflexive symmetric*. If for every pair $(\rho, \theta) \in S$ there exists a pair $(\rho, \pi + \theta) \in S$ then $\vec{C}$ is π -rotationally symmetric. Hereafter, π- rotational symmetry is often referred to as rotational symmetry.

3.2 Skew Formulations

Skew symmetric figures are planar figures in the image space that are generated from planar symmetric figures by one of two processes :

(1) rotation in three dimensions followed by orthographic projection onto the image plane;

(2) oblique coordinate transformation followed by rotation in the image plane.

Under projection, information about the underlying symmetric shape and its orientation are lost. Thus, skew symmetric figures are ambiguous since there are many figure-transformation pairs capable of generating a given skew symmetric figure. This loss of information is the result of two distinct types of ambiguity : *gradient* ambiguity and *axial* ambiguity.

Gradient ambiguity exists in the process of projection onto the image plane because there exists an infinite number of rotations that project the x and y axes orthographically to the same pair of skew symmetric axes. The solutions for the plane orientation are constrained to lie along the locus of a hyperbola in the gradient space [kan81]. *Axial* ambiguity is defined as the number of skew symmetry axes, i.e., degree of skew symmetry, of the skew symmetric contour. (This defintion is not equivalent to the definiton for *shear ambiguity* given in [freid86], see [gros90a].) This definition of axial ambiguity is consistent with a view towards recovering the underlying local 2D coordinate system of the planar figure in the scene. A way of viewing axial ambiguity is as the number of local oblique coordinate systems that can be selected given a skewed image contour. This ambiguity is directly computable from the image.

In this paper, the oblique-coordinate-transformation formulation of skew is used (as described above) rather than the general one of 3D orthographic mapping. The motivation for this is basically the same reason as that given in [freid86], i.e., it is mathematically convienent. The mapping from 3D orthographic projection to image skew tranformation is, unfortunately, infinitely many to one. It is many to one, generally refered to as gradient ambiguity, because uncountably many different {planar figure, 3D orientation} pairs project onto the same image contour. By contrast, the mapping from orthogonal image coordinates to oblique image coordinates is one to one.

To skew a symmetric image in the plane, a skew coordinate transformation is required. The skew transformation is given by $Tp = q$, where

$$T = \begin{bmatrix} \cos\alpha & \cos\alpha\tan\beta - \sin\alpha \\ \sin\alpha & \sin\alpha\tan\beta + \cos\alpha \end{bmatrix} \tag{3.2.1}$$

Transformation T in effect maps an 2D orthogonal coordinate system into a rotated oblique one. The angle β is the degree of skew, where no skew ($\beta = 0$) corresponds to a purely orthogonal coordinate system under rotation. The α angle rotates the oblique coordinate system in the image space.

Each pair of skew axes that is found corresponds to a skew coordinate transformation from a symmetric figure into the given image skew symmetric contour. Where multiple axes of skewed symmetry exist, SYMAN attempts to find all such solutions (see section 4).

The inverse transformation T^{-1} is useful in *deskewing* the image contour, and is given by

$$T^{-1} = \begin{bmatrix} \cos\alpha + \sin\alpha\tan\beta & \sin\alpha - \cos\alpha\tan\beta \\ -sin\ \alpha & \cos\alpha \end{bmatrix} \tag{3.2.2}$$

In section 1, an informal definition of skew symmetry was given. Using the inverse skew transformation T^{-1} given in equation (3.2.2), we can define skew symmetry formally.

Let $\vec{C} \in R^2$ be a contour in the image plane. Then contour $\vec{C}$ is *skew* symmetric *iff* there exist angles α_0 and β_0 such that, for every point $p_1 \in \vec{C}$ there exists a point $p_2 \in \vec{C}$ such that

$$P_x\, T^{-1} p_1 = -P_x\, T^{-1} p_2\,,\ P_y\, T^{-1} p_1 = P_y\, T^{-1} p_2\,, \tag{3.2.3}$$

where P is the projection function.

As stated in section 1, this paper presumes as input to the shape recovery algorithm a set of skew symmetric contours. SYMAN, described in section 4, is capable of solving for axes of skew *given* a skewed contour. Although no algorithm is presented in this paper for *finding* skewed contours in an image (but see [gros90a] for some moment-based techniques), the constraints given in (3.2.3) provide a necessary and sufficient condition for a contour to be skew symmetric (although it does not allow for occlusion).

3.3 Some Propositions

From the definitions in section 3.1, some propositions can be derived:

Proposition 3.1: If a curve $\vec{C}$ is doubly reflexive symmetric it is also π-rotationally symmetric.

Proof: Since $\vec{C}$ is doubly reflexive symmetric, for every point $(x, y) \in \vec{C}$ there exist points $(-x, y) \in \vec{C}$ and $(x, -y) \in \vec{C}$. But if $(x, -y) \in \vec{C}$ and $\vec{C}$ is reflexive symmetric, then $(-x, -y) \in \vec{C}$.

Proposition 3.2: The property of π-rotational symmetry is invariant to the affine transformations of skew, rotation, and scaling.

Proof: Let T be a transformation matrix that skews, rotates, and rescales a point p into a point q. We can write this as $T\,p = q$. But then $T\,(-p) = -(T\,p) = -q$.

From Propositions 3.1 and 3.2 and the relationship between oblique coordinate transformation and 3D rotation followed by orthographic projection (section 3.2), we have the following corollary:

Corollary: If a planar 3D scene curve C is doubly reflexive symmetric then its 2D orthographic image projection C_I is π-rotationally symmetric.

Of course, the converse of Proposition 3.1 is not true. A curve can be π-rotationally symmetric but not doubly reflexive symmetric. An example of this is the sine function, which is rotationally symmetric about the origin but is not even singly reflexive symmetric, let alone doubly reflexive symmetric. The curve in figure 1a is doubly reflexive symmetric, while its projection (i.e., skewed contour), shown in figure 1b, is no longer doubly reflexive symmetric nor even singly reflexive symmetric. It is skew symmetric, however, and also π-rotationally symmetric. Generally, due to the compact nature of real-world objects, rotationally symmetric contours in the image plane are the projections of doubly reflexive symmetric shapes in the scene. This motivates a heuristic used by SYMAN in analyzing skewed symmetry. The heuristic, called the *rotational symmetry heuristic*, is stated as follows:

> "If a contour in the image is π-rotationally symmetric and can be interpreted as the projection of a doubly reflexive symmetric curve in the scene (i.e., has at least two skew-symmetric axes), then it is presumed to be the projection of a doubly reflexive symmetric scene curve."

4.0 SYMAN

Section 4.1 describes SYMAN, a system constructed by the author for the purpose of analyzing skewed symmetries. As SYMAN is not the focus of this paper, but rather a module in a larger 3D shape recovery method, the description of the system is necessarily brief. For a more detailed description of SYMAN see [gros90a]. Section 4.2 provides several examples of input to SYMAN and the solutions generated by the system. This should give the reader a clearer understanding of what has been heretofore referred to in this paper as axial ambiguity.

4.1 A description of SYMAN

In this section, the symmetry analyzer SYMAN is briefly described. The motivation behind SYMAN was the fact the the universe is, by and large, a very regular, symmetric place. As such, a robust 3D shape recognition system has a need for a module that understands a lot about symmetries and their projections. Although there have been other algorithms derived for finding axes of skew given a skewed contour, none of them were general enough in scope to be termed a skew analyzer, capable of tasks such as conjecturing the degree of skew ambiguity. A survey of current techniques for finding axes of skew is given in [gros90b].

SYMAN consists of different components. In particular, it is capable of running 4 different algorithms to recover axes of skew. Two of these algorithms are derived in [gros90a]. One of these algorithms is a fast, moment-based technique, robust under noise and discontinuities, but that does not perform well under occlusion. The second algorithm is local, i.e., required to make pointwise comparisons to look for skew pairs. It is based on a constraint derived between tangent values of skew pairs of points and is able to solve for axes of skew under occlusion. The other two methods implemented as part of SYMAN are adapted from [ponce90] and [marc90]. The method described in [ponce90] has a necessary curvature-based constraint among pairs of skew symmetric contour points. This necessary constraint is used analogously to Brady's technique for finding axes of smooth local symmetry [brad84]. The method in [marc90] uses a constraint (to solve for α of transformation T, equation 3.2.1) that requires a pure skew (no rotational component) to have a linear bisector of the local transverse axes (lines connecting corresponding skew points). The reason for wanting SYMAN to implement all of these methods is because each of the methods appears to have certain strengths and the trick seems to be in knowing when to use each technique (see [gros90b]). For example, the moment-based method used by SYMAN does not perform well under partial occlusion or when the degree of skew symmetry is large (greater than 4). The method of [ponce90] does not work well on linear or discontinuous contours. The line-based method of [marc90] is very sensitive to outliers, which occur when contours are complex and non-convex. By having several methods to choose from, and some decision criteria for when to use each, it is hoped that SYMAN can be made more robust. At the current time, SYMAN relies primarily on the moment-based method and the local tangent-based method.

Aside from being able to recover axes of skew, SYMAN also tries to *understand* the contour. It tests for whether the contour is a real symmetry under rotation, i.e., no skew component. It checks if the contour is a rotational symmetry. When a solution is recovered, it tests whether it is valid by seeing if certain moments in the recovered contour are approximately equal to zero.

One of the more important features of SYMAN is the realization that there are some things a symmertry analyzer should *not* resolve. Attached to each contour is a structure. If SYMAN determines that the contour is rotationally symmetric, it will find 4 pairs of skew axes. Rather than attempting to decide which of these is correct, as several other algorithms do, SYMAN enters these solutions, as well as a conjecture that the contour is the projection of a doubly symmetric figure into the contour's data structure. Since in this case there really are 4 such pairs of skew axes, it makes sense not to try and resolve the *axial ambiguity* on this level, but rather to pass it on to a higher level in the shape recovery algorithm that is more *context-sensitive* and can hopefully arrive at a unique set of skew axes using constraints from adjacent contours (as described in the next section).

Realizing that moments are fast but subject to bias, SYMAN has a bias correcting module. Given a solution, which it assumes is *close* to being correct, it polishes the solution to greater accuracy using a measure that is more insensitive to noise and discretization effects. This bias-correcting module substantially improves the accuracy of recovered solutions. SYMAN is intended to function as a module in a larger 3D shape recovery system that is capable of recovering 3D shape from monocular intensity images by assuming an underlying model of symmetry.

The moment-based algorithm used by SYMAN is given (in pseudo-code) below:

1. Compute moments (up to fourth order) of the image contour and normalize the moments with respect to the contour centroid.

2. Solve for the rotation parameter α assuming the contour is a pure rotation. Having solved for α, test if the contour is really a pure rotation.

3. Test if the contour is π-rotationally symmetric.

4. If (contour is π-rotationally symmetric) {indicative of an underlying 4th order real symmetry in the scene}

```
        use 4th degree skew function ;
    else
        use 3rd degree skew function ;
```

5. Check if the solutions are valid.

4.2 Some Examples Using SYMAN

In this section some of the solutions recovered by SYMAN for axes of skewed symmetry are shown for both real and synthetic iamges.

Consider the skewed and rotated superellipse of figure 5a. SYMAN solved for four pairs of (α, β) solutions. Of course, of the 4 solutions only 2 are unique; the other two are duals of the first two, with the skew and transverse axes reversed. There are two ways to display solutions. One is to draw the recovered axes of skew on top of the skewed image contour; the other is to actually derotate and deskew the image contour. We will alternate between these two methods.† Two solutions for the skew symmetric contour of figure 5a are shown (in deskewed, derotated format) in figure 5b and figure 5c.

Consider the skewed picture, taken from Mohan [moh89] (originally from Marr), shown in figure 6a. There are 4 solutions recovered by SYMAN (two solutions are duals of the first two), two of which are shown in figures 6b and 6c. Next, SYMAN's moment-based algorithm algorithm was run on the die image shown in figure 7 (the image is due to Glenn Healey). The edge contours shown were detected with a Canny-like edge detector then labeled using a connected-component algorithm. The recovered skew axes are shown overlaid on top of the original intensity image. None of the skewed contours in the die image had more than one solution for the axis of skew (i.e., there was no axial ambiguity).

The edge contour of a leaf image is shown in figure 8a. Of the three solutions recovered by SYMAN, one is shown (the one that is closest to the correct solution) in figure 8b. An edge contour of a image taken of a symmetric ink blot is shown in figure 9a. A solution recovered by the SYMAN moment-based algorithm is shown in figure 9b.

The axial ambiguity previously described in this paper is clearly in evidence for figures 5 and 6. A context-sensitive method for resolving such axial ambiguity is described in the next section.

† The derotated and deskewed contour should not be confused with the real 3D scene contour. There is still gradient ambiguity here that cannot be resolved without additional context or heuristics.

5.0 Resolving Axial Ambiguity

Skewed image contours intrinsically contain information about local coordinate systems, namely, the skew symmetry and skew transverse axes (see section 1). Often, a given skew contour has axial ambiguity, which SYMAN is capable of detecting. Because resolving axial ambiguity is a context-sensitive problem, SYMAN makes no attempt to resolve such ambiguities, but rather stores all the recoverd solutions for axes of skew in a data structure associated with the skew curve. This section develops a method for resolving axial ambiguity given a set of adjacent contours and their respective sets of local coordinate systems.

To find a unique local coordinate system for each skew contour, adjacent contours are considered. This can be clarified with an example. Consider the intensity image of a Ferrari model car shown in figure 10. The edge map recovered from a Canny-like edge detector is shown in Figure 11. A group of hand-selected adjacent edges is shown in Figure 12. The selection of adjacent contours was done by hand in this example, though this seems like a task that can be automated. When these contours were given to SYMAN as input, there were two distinct solutions generated for each of the 4 skew curves. These recovered solutions for axes of skew are shown in figures 13a, 13b, 14a, 14b, 15a, 15b, 16a, and 16b.

Can we find a unique set of coordinate axes for each skew symmetric contour in this set of adjacent contours? We observe that for each skew contour there is one oblique coordinate system solution such that one of the coordinate axes is parallel to one of the coordinate axes at each of the other adjacent contour's oblique coordinate axes. Why is this a desirable property, namely, what do coincident axial directions of adjacent surface patches tell us about the underlying regularity of the surface? Consider figure 17, taken from [stev81]. It is the image of an orthographically projected *developable, (singly curved,) locally cylindrical* surface. Stevens notes that the minimal line of curvature remains parallel as one moves along the surface's maximal line of curvature. Thus, if a set of adjacent local coordinate systems all have a parallel coordinate axis direction, using a non-accidentalness criterion analogous to [lowe85], it is strongly indicative of an underlying locally cylindrical 3D surface in the scene. As in [lowe85] and [moh89], parallel cues are within a certain tolerance; in the examples in this section, lines were considered parallel if they were within ± 15 degrees. For the four contours in figure 12, the solutions for the skew transverse axes (i.e., the rotational component of skew transformation T), were sorted. There was a cluster detected at approximately -18.5 degrees (with respect to the horizontal image axis); each of the skewed contours had a transverse axis solution within three degrees of -18.5 degrees. This sorting technique among the transverse axes of adjacent contours (which determine the rotational component of the skew) followed by a search for clusters of solutions was used as a filter to resolve axial ambiguity. The criterion can be relaxed to find a consistent set of local coordinate axes among *neighboring* skew contours, that are not strictly adjacent. Using this criterion, a unique local oblique 2D coordinate system was recovered for each of the skew axes in figure 12. The skewed contours and recovered axes are shown overlaid on the car intensity image in figure 18. Note that the elliptical tire hubcap has been included as it supports the same parallel axis, although it is not strictly adjacent to the other contours.

Resolving axial ambiguity is generally a very local operation. In fact, two adjacent contours are generally sufficient to uniquely solve for the axes of skew at both contours. Consider two skew symmetric contours from figure 12 juxtaposed, as shown in figure 19. While each one has axial ambiguity, as can be seen from figures 15 and 16, when considered together there is no ambiguity as to the local oblique coordinate systems for each region; they are the coordinate systems recovered by SYMAN and shown in figures 15a and 16a.

Next, consider the set of neighboring contours shown in figure 20 that were hand-selected from the edge image in figure 11. Here there is also some degree of axial ambiguity. The top of the car, for example, was already shown to have two solutions for its axes of skew (figure 16). Another contour from this set, the car window, is shown with its two different sets of axes recovered by SYMAN in figures 21a and 21b. By sorting the set of coordinate solutions recovered by SYMAN for the contours in figure 20 and then finding a cluster of approximately equal solutions for the transverse axis (rotation parameter α), the axial ambiguity was resolved. Each of the contours in figure 20 was found to have a solution for the skew transverse axis of approximately 43 degrees (± 3 degrees). The resolved sets of skew axes for the edges of figure 20 are shown in figure 22 overlaid on the original car intensity image. There is clear support among this set of recovered local coordinate systems for a particular axial direction.

The parallel direction supported by the local coordinate systems in figure 22 (analogous to figure 18) appears to be the projection of an underlying axis of the 3D surface. The relevance of this projected axis to the underlying object-centered coordinate system is studied in section 7. It is simply noted here that some of this support comes from contours that by themselves do not lend support to a particular axial direction. Consider the front wheel well of the car. The curved contour of the wheel well does not support a particular axial direction. It is only by solving for the transverse skew axis of the contour that a meaningful support axis is recovered.

Applying the same technique on yet another set of edges from figure 11, (i.e., running SYMAN and then filtering the set of solutions to recover unique local coordinate systems), another set of skew symmetric axes is recovered, as shown in figure 23.

This section developed a method of resolving axial ambiguity for classes of regular surfaces that are locally cylindrical and have symmetric regions. The implicit heuristic used here was that local coordinate systems at adjacent contours are, in general, not arbitrarily aligned with respect to each other. In particular, if a set of neighboring skew symmetric contours are the projection of a set of symmetric 3D curves sitting on an approximately locally cylindrical surface, then they have an approximately parallel coordinate axis that can be used to resolve axial ambiguity. Since many doubly curved surfaces can be approximated locally by singly curved, cylindrical surface patches, this heuristic is actually quite general.

6.0 Constraining gradient ambiguity

In the previous section, a heuristic based on symmetry and nonaccidental alignment was used as a filter to recover local coordinate axes uniquely. Resolution of axial ambiguity is not extremely context-sensitive. As was shown in figure 19, it is generally sufficient to compare two adjacent contours in order to resolve axial ambiguity. The problem of resolving gradient ambiguity, on the other hand, is inherently more context-sensitive. While a skewed contour usually has a very finite degree of axial ambiguity, a given pair of skew symmetric axes have an infinite degree of gradient ambiguity, i.e., a hyperbola in the gradient space. For a set of adjacent skew symmetric contours, there is generally no method for recovering the unique gradient for each skew symmetric region. Using a method from Stevens [stev81], however, the gradient solution at each contour can be constrained. In this section, Stevens' method is reviewed briefly and then applied to constrain the gradient value of contours in the car image (figure 11). In section 7, an image basis is found for the car which allows us to solve for the gradient at skew symmetric contours uniquely.

To understand how local pairs of skew axes on a cylindrical surface can be used to constrain the gradient space solution at each of these local surface patches, some results from Stevens are briefly reviewed here. Consider a locally cylindrical surface S and a point $P_1 \in S$. At the point P_1 there is a local orthogonal coordinate system whose axes are parallel to a pair of vectors U and V in R^3. For convenience, they are constructed to project as unit vectors and, without loss of generality, the x-axis is oriented with the projection U. Let β_1 be the measured angle between the projections of U and V. Then the U and V vectors are given by

$$U = \{1, 0, a\} \quad , \quad V = \{\cos\beta_1, \sin\beta_1, b\} \quad . \tag{6.1}$$

The quantities a and b are the unknown components of U and V along the z-axis (perpendicular to the image plane). The surface normal N is just the vector product $U \; X \; V$, given by:

$$N_1 = \{-a \sin\beta_1, a \cos\beta_1 - b, \sin\beta_1\} \tag{6.2}$$

and tilt τ and the slant σ are :

$$\tau = \tan^{-1}\frac{N_y}{N_x}, \quad \sigma = \cos^{-1}\frac{N_z}{(N^2{}_x + N^2{}_y + N^2{}_z)^{1/2}} \tag{6.3}$$

The expression for the normal in equation (6.2) has two unknowns. Since U and V are orthogonal in R^3, the constraint $U \cdot V = 0$ can be applied to the expressions for U and V (6.1) to obtain

$$b = -\frac{\cos\beta_1}{a} \tag{6.4}$$

which when substituted into equation (6.2) yields

$$N_1 = \{-a \sin\beta_{1,} \cos\beta_1 \frac{a^2+1}{a}, \sin\beta_1\}. \tag{6.5}$$

This constrains the surface orientation to a one parameter space, a. This degree of freedom is equivalent to Kanade's hyperbola in the gradient space (see [stev81]). The angle of intersection β_1 is computable from the image axes. Substituting the expressions for N_x and N_y from equation (6.5) into the expression for τ (6.3) and solving for a, we have that

$$a = \left(\frac{-1}{\tan\tau \tan\beta_1 + 1} \right)^{1/2} \tag{6.6}$$

Next, consider a second point $P_2 \in S$. The point P_2 has a local orthogonal coordinate system whose axial directions are vectors parallel to U and W in R^3, see Figure 24. The local orthogonal coordinate system at P_2 projects onto a pair of skew axes in the image that intersect at an angle of β_2. The surface normal at P_2 is given by

$$N_2 = \{-a \sin\beta_{2,} \cos\beta_2 \frac{a^2+1}{a}, \sin\beta_2\}. \tag{6.7}$$

Since N_1 and N_2 are both functions of unknown parameter a, a constraint or solution of one normal value will propagate to the other normal value (see [stev81]).

For a given angle β to correspond to the projection of a perpendicular intersection, the intersection can only have certain 3D orientations with respect to the viewer. These limits are shown in Figure 25. Observe that if β is near π, the foreshortening of the 3D intersection must be large (Figure 25a), but if β is near $\frac{1}{2}\pi$ one can say little about surface orientation. This restriction has a simple interpretation in terms of tilt, see Figure 25b. If the tangents to the curves in the image plane have orientations τ_1 ans τ_2 at their intersection, then the surface tilt τ (i.e., projection of the surface normal) must lie within the perpendiculars to these tangents :

$$\tau_1 + \frac{1}{2}\pi \le \tau \le \tau_2 - \frac{1}{2}\pi . \tag{6.8}$$

Stevens suggests that as an estimate of tilt parameter τ, the bisector of this interval be selected. Thus, given a set of adjacent contours having a mutually parallel coordinate axis, equations (6.7) and (6.8) can be used to estimate the surface normal at each of the skew contours. The method first finds the contour whose skew symmetric axes have the largest angle of intersection β. Using equation (6.8), the bisector of the interval in which τ must lie is taken as an estimate for the tilt at that contour. But this allows us to estimate a in equation (6.5), which in turn gives us an estimate of the 3D surface normal. But having computed an estimate for a, the surface normals of the other skew symmetric contours can also be estimated using equation (6.7). Using Stevens' constraint, we can constrain the tilt τ to lie within an interval. In general, however, there is no way to recover the tilt or the surface normal more precisely.

Consider the set of skew symmetric contours in Figure 12. The contour with the largest angle of intersection is the contour in figure 14a, where SYMAN computed a value of $\beta = 59.9$ degrees, where $\tau_1 = 23.275$ and $\tau_2 = 161.675$. From equation (6.7) we have

$$113.275 \le \tau \le 71.675$$

Since this skew symmetric contour has the largest angle of intersection, it also has the smallest interval in which τ can lie. Nevertheless, for this example, the interval in which it can lie is 41.6 degrees. The bisector would estimate τ as being approximately 92.5 degrees. The curve together with the tilt interval in which τ is constrained to lie is shown in Figure 26. The bisector of 92.5 degrees is quite different than the much more accurate estimate derived after computing an object basis (section 7). The tilt value computed for this curve using the method described in the next section was 79.1 degrees (see Figure 29). Thus, even though we can constrain the gradient space solution of the normal to a symmetric surface region, we cannot solve for it uniquely. In addition, the tilt interval may be quite large. The next section describes a method for more accurately estimating the normal (alternatively the tilt) for (approximately) symmetric surface regions.

7.0 Finding the Projected Object Basis

Most object parts in the real world can be parameterized with respect to a 3D orthogonal coordinate system. This is true for shapes such as conic surfaces, straight generalized cylinders and superquadrics. The object-centered coordinate system needs to be recovered in order to match the recovered shape to models in a database. In addition, if the projected set of object-centered coordinate axes can be determined from the image directly then the transformation between viewer-centered and object-centered coordinate systems is easily computed (see [barn84]) to a mirror reversal. But for symmetric objects, like the car in figure 10, there is a natural 3D orthogonal coordinate system in which to parameterize it and the object's image basis (set of projected coordinate axes) can often be determined directly from the image.

Consider the recovered skew symmetric axes in Figure 18. There is an axial direction shared by all the skewed contours. The fact that all the contours have a common axial direction is viewed as an indication that that direction is significant with respect to the underlying coordinate system. This is also true of the dominant axes in Figures 22 and 23. The skew symmetric contours shown in Figure 27 were given as input to SYMAN. All the solutions returned by SYMAN were sorted and the mean of the three most prominent clusters were selected as the set of projected image axes. The detected image axes P_i, P_j and P_k intersect the horizontal axis at angles of -137, -18.5, and 88 degrees. They are shown, overlaid on the car intensity image, in figure 28. Having the image axes, the underlying 3D coordinate system is easily computed (see [barn84]) and are given in (x, y, z) viewer-centered coordinates (where positive x is the right horizontal image axis, positive y is the upper vertical image axis, and positive z is perpendicular to the image plane, coming out of the page) by

$$\vec{i} = -.5\,\vec{x} - .451\,\vec{y} + .735\,\vec{z}$$
$$\vec{j} = .863\,\vec{x} - .304\,\vec{y} + .4\,\vec{z}$$
$$\vec{k} = .0435\,\vec{x} + .836\,\vec{y} + .544\,\vec{z}$$

Having solved for the object centered coordinate system, the 3D normals at skew symmetric contours that have an axis parallel to one of the object-centered axes can now be solved for uniquely. For example, the top of the car shown in Figure 16a had recovered image axes of -18.7 degrees and -136.9 degrees. Since the first image axis at -18.7 degrees is parallel (approximately) to the object-centered $\vec{j}$ direction, we need only to find the other axis in viewer-centered R^3. Let the other 3D axis of the contour be given by $\vec{u}$. Since the skew symmetric axes are orthogonal in 3D, we have $\vec{j} \cdot \vec{u} = 0$. Since P_u, the projection of $\vec{u}$ in the image, intersects the horizontal axis $\vec{x}$ at an angle of -136.9 degrees, a non-unit magnitude $\vec{u}$ can be written as

$$\vec{u} = -.731\,\vec{x} - .682\,\vec{y} + a\,\vec{z}$$

Solving for a we obtain $a = 1.05$. The unit magnitude vector is given by

$$\vec{u} = -.504\,\vec{x} + -.47\,\vec{y} + .724\,\vec{z}$$

Note that $\vec{u}$ is approximately parallel to object coordinate axis $\vec{i}$ as one would have expected. The unit normal to the surface, obtained from the vector product of $\vec{j}\ x\ \vec{u}$, is given by

$$N_C = .032\,\vec{x} + .825\,\vec{y} + .55\,\vec{z}$$

For the tilt, which is just $\tan^{-1}\frac{N_y}{N_x}$, the projected normal vector, we have $\tau_C = \tan^{-1}(2.223) = 87.78$ degrees. Similarly, the surface normal vector can be computed for each symmetric surface region that projects onto one of the detected skew symmetric contours in Figures 18, 22, and 23. The car image with some of the computed surface normals, displayed by showing the projected normal (tilt) at each skew symmetric contour, is shown in Figure 29.

The section, then, has described a method for recovering surface normals for an object that is (to an approximation) locally cylindrical with symmetric surface regions by determining a natural object basis. After the projected object-centered coordinate axes have been recovered, the position of these axes in viewer-centered R^3 can be uniquely computed (up to mirror reversal). But if the 3D object-centered coordinate system can be recovered, this can then be used to solve for the 3D coordinate systems at symmetric surface regions that have an image axis that is approximately parallel to one of the projected object-centered axes. This in turn gives us precise surface normals that could only be constrained using Stevens's surface-based gradient constraint.

8.0 Conclusion

This paper developed a method for recovering the shape of object surfaces that can be approximated locally by cylindrical surfaces and that contain symmetric regions. A dichotomy was made between two different types of amiguity that occur frequently in analyzing the projections of symmetries: gradient ambiguity and axial ambiguity. It was shown that while resolving either of these classes of ambiguity is a context-sensitive operation, gradient ambiguity is generally more context-sensitive and harder to resolve than axial ambiguity.

Axial ambiguity, the ambiguity as to which coordinate system among a set of possible local oblique coordinate systems is correct, can be resolved by considering adjacent contours. For locally cylindrical surfaces, adjacent contours have at least one mutually parallel axis. Once a unique coordinate system has been recovered at each skew symmetric contour, the gradient value of the symmetric region needs to be recovered. Using a constraint from Stevens, the tilt (as well as the surface normal) can be constrained to lie within some interval. The gradient value of a symmetric region, however, cannot be uniquely recovered using the Stevens constraint.

The projected object basis of a symmetric regular object (a car model) was determined in section 7 by filtering the solutions for axes of skew symmetry recovered by SYMAN and looking for clusters of axial directions that were dominant. Having recovered the image object basis, the transformation between viewer- and object- centered coordinate systems was solved for uniquely. The recovered 3D object-centered coordinate system was used, in turn, to uniquely solve for the surface normals of symmetric regions on the object surface.

This work was primarily concerned with recovering shape from objects that are symmetric and locally cylindrical. Future work includes finding ways to generalize some of the techniques used on singly-curved surfaces to doubly-curved surfaces. In addition, there is a need for robust methods that automatically detect image regions corresponding to the projections of symmetric regions on the surface.

References

[barn84] Barnard, S., Choosing a basis for perceptual space, *IEEE Workshop on Computer Vision: Representation and Control*, 1984.

[brad84] Brady, J.M., and Asada, H., Smoothed local symmetries and their implementation, *International Journal of Robotics Research 3*, No. 3, 1984.

[brad83] Brady, J.M., and Yuille, A., An extremum principle for shape from contour, MIT, AI Lab., MIT-AIM 711, 1983.

[brook81] Brooks<, R. A., Symbolic reasoning among 3-D models and 2-D images, *Artificial Intelligence* **17**, 285-348, 1981.

[freid86] Freidberg, S.A., Finding Axes of Skewed Symmetry, in *Computer Vision, Graphics, and Image Processing*, **34**, 138-155, 1986.

[gros90a] Gross, A.D., Boult, T.E., Analyzing Skewed Symmetries, Technical Report CUCS-064-90, Department of Computer Science, Columbia University.

[gros90b] Gross, A.D., Boult, T.E., Skew symmetry detection: Evaluating the state of the art, Technical Report, Department of Computer Science, Columbia University.

[hor87] Horaud, R., and Brady, J.M., On the geometric interpretation of image contours, in *Proc. First Int'l Conference on Computer Vision*, London, U.K., June 1987.

[kan81] Kanade, T., Recovery of the 3D shape of an object from a single view, *Artificial Intelligence*, vol. 17, pages. 409-460, 1981.

[lowe85] Lowe, D. G., *Perceptual organization and visual recognition*, Kluwer Academic Publishers, Boston, 1985.

[moh89] Mohan, R., Perceptual organization for computer vision, PhD thesis, School of Engineering, University of Southern California, August, 1989.

[nal87a] Nalwa, Vic, Line-drawing interpretation: Bilateral symmetry, in *Proceedings of the Image Understanding Workshop*, vol. 2, Feb., 1987, pp. 956-967.

[nal87b] Nalwa, Vic, Line-drawing interpretation: Straight-lines and conic-sections, in *Proceedings of the Image Understanding Workshop*, vol. 2, Feb., 1987, pp. 942-955.

[ponce90] Ponce, J., On characterizing ribbons and finding skewed symmetries, accepted for publication in *Computer Vision, Graphics, and Image Processing*, Fall 1990.

[ponce89] Ponce, J., and Kriegman, D.J., On recognizing and positioning curved 3D objects from image contours, in *IEEE Proceedings of the Workshop on 3D Interpretation of Scenes*, pages 116-123, Nov. 27-29, 1989, Austin, TX.

[rao88] Rao, K., Shape description from sparse and imperfect data, Technical Report IRIS-250, Institute for Robotics and Intelligent Systems, USC, December 1988. Ph.D. Thesis.

[rosen86] Rosenfeld, A., Axial representation of shape, *Journal of Computer Vision, Graphics, and Image Processing*, (33):156-173, 1986.

[marc90] Saint-Marc, P., and Medinoni, G., B-spline contour representation and symmetry detection, technical report, Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles, CA, Feb 20, 1990.

[stev81] Stevens, K.A., The visual interpretation of surface contours, *Artificial Intelligence***17**, 47-73, 1981.

[ulup88] Ulupinar, F., and Nevatia, R., Using symmetries for analysis of shape from contour, in *Proceedings of the Second International Computer Vision Conference*, 414-426, Dec. 5-8, 1988.

[yale88] Yale, P., *Geometry and Symmetry*, Dover Books, San Francisco, 1988.