On Pencils of Tangent Planes and the Recognition of Smooth 3D Shapes from Silhouettes

Svetlana Lazebnik[1], Amit Sethi[1], Cordelia Schmid[2], David Kriegman[1], Jean Ponce[1], and Martial Hebert[3]

[1] Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
405 North Mathews Avenue
Urbana, IL 61801 USA
{slazebni, asethi, kriegman, j-ponce}@uiuc.edu
[2] INRIA Rhône-Alpes
665, avenue de l'Europe
38330 Montbonnot, FRANCE
Cordelia.Schmid@inrialpes.fr
[3] Carnegie Mellon University Robotics Institute
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
hebert@ri.cmu.edu

Abstract. This paper presents a geometric approach to recognizing smooth objects from their outlines. We define a signature function that associates feature vectors with objects and baselines connecting pairs of possible viewpoints. Feature vectors, which can be projective, affine, or Euclidean, are computed using the planes that pass through a fixed baseline and are also tangent to the object's surface. In the proposed framework, matching a test outline to a set of training outlines is equivalent to finding intersections in feature space between the images of the training and the test signature functions. The paper presents experimental results for the case of internally calibrated perspective cameras, where the feature vectors are angles between epipolar tangent planes.

1 Introduction

Many recognition systems represent objects using features derived directly from image intensity patterns. These systems work well on textured animals [10] or objects with distinctive markings, like faces and cars [11,12]. However, they are limited in their ability to distinguish between objects based on true 3D shape. They may fail, for instance, to tell the difference between a tiger and a tiger-skin rug. Another difficulty is that some classes of objects do not have intensity or color descriptors with sufficient discriminative power: in the absence of surface texture or markings, the silhouette becomes the main clue to the object's identity.

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2352, pp. 651–665, 2002.

A common approach to silhouette-based matching consists of finding rich local descriptors for a set of contour points. This process may involve computing orientation information associated with pairs and triples of points [3] or attaching two-dimensional "shape context" histograms to each point [2]. An important advantage of these methods is that they do not require complete segmentation, working instead with a scattered set of edge points. However, most known contour descriptors are mainly suitable for 2D recognition — from a geometric point of view, it is hard to justify the appropriateness of arbitrary outline statistics for matching multiple views of the same 3D object.

In this paper, we present a true geometric approach to recognizing smooth 3D objects. We follow the general philosophy of deriving a rich silhouette description to build a highly descriptive feature space, while taking care to define features that have a rigorous 3D interpretation. In our framework, a potential match between two outlines is a hypothesis of a consistent epipolar geometry between the two respective viewpoints. Previous work on geometric silhouette matching has been limited, considering only weak perspective or restricting the set of allowable camera movements [1,7,9]. The approach proposed in this paper is fully general, encompassing the cases of uncalibrated and internally calibrated perspective projection, as well as affine projection. Our method is not restricted to outlines taken from nearby viewpoints, and explicitly accounts for self-occlusion.

In the following section, we give a conceptual introduction to our recognition framework. In Sections 3 and 4, we discuss the properties of the feature space and the conditions for matching outlines. In Section 5, we address implementation issues and report results from a preliminary recognition experiment.

2 Recognition Framework

Assume that we are given a training set of outlines of a single object. We want to construct a representation suitable for recognizing instances of this object based on outlines in test images from viewpoints not present in the training set. The key idea is the following: for each image, we associate a set of invariants with each possible *baseline* connecting its viewpoint to any other viewpoint. The baseline between two camera centers determines the epipolar geometry of the scene: the *epipoles* are the intersections of the baseline with the image planes, and the *epipolar lines* are intersections of the image planes with the pencil of *epipolar planes* passing through the baseline (Figure 2). The epipolar geometry does not change when we translate the camera centers while keeping the baseline fixed. We take advantage of this invariance property by introducing a *signature function* $\mathcal{S}$ that assigns a vector from some *feature space* $\mathcal{F}$ to each possible baseline. The feature vectors (to be defined precisely in Section 3.3) can be computed *in the image* given an outline and a hypothesized epipole, but they measure properties of the 3D object *in space*.

Let Γ denote the collection of the outlines in the training images (a discrete set of pictures or a video clip). Any point $\mathbf{e}$ on the projective image plane of some γ in Γ is a potential epipole, and is identified with a unique line (the line

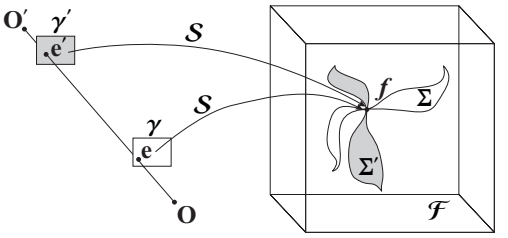


Fig. 1. Outlines γ and γ' are connected in space by the baseline through their respective camera centers, $\mathbf{O}$ and $\mathbf{O}'$. This baseline intersects the image planes in epipoles $\mathbf{e}$ and $\mathbf{e}'$. Computing signature functions $\mathcal{S}(\gamma, \mathbf{e})$ and $\mathcal{S}(\gamma', \mathbf{e}')$ yields a vector f in feature space $\mathcal{F}$ that lies on the intersection of the two signature surfaces Σ and Σ'.

passing through the camera center and piercing the image plane at the epipole location). Thus, sampling all possible epipoles is equivalent to sampling the two-dimensional set of all baselines passing through the camera center. Indeed, the space of all lines through the origin in 3-space is topologically equivalent to $\mathbb{P}^2$, the projective plane. The signature function $\mathcal{S}$ encapsulates the relationship between outlines and epipoles/viewing rays as follows:

$$\begin{aligned} \mathcal{S} : \Gamma \times \mathbb{P}^2 &\to \mathcal{F} \\ (\gamma, \mathbf{e}) &\mapsto f \ . \end{aligned}$$

Since the feature space $\mathcal{F}$ contains information about the 3D properties of the scene, it is actually possible to express $\mathcal{S}$ as a function of an object and a line in space. However, the definition above emphasizes the 2D information that is directly accessible to the recognition algorithm, namely, an outline and a point on the image plane.

Let γ and γ' denote a training and a test outline of the same object (keep in mind that the relative camera positions are unknown). Consider the baseline connecting the camera centers of γ and γ'. This baseline yields a pair of epipoles: $\mathbf{e}$ in the image plane of γ, and $\mathbf{e}'$ in the image plane of γ'. Since the training and the test images capture the same object and the two epipoles refer to the same line in space, we must have

$$\mathcal{S}(\gamma, \mathbf{e}) = \mathcal{S}(\gamma', \mathbf{e}') \ .$$

Now, suppose that this equality holds for some particular γ, γ', $\mathbf{e}$, and $\mathbf{e}'$. Then the two epipole positions define a baseline for which the two pictured outlines are consistent with a single object. If the feature space $\mathcal{F}$ is sufficiently high-dimensional, then a match of signature functions is a strong indication that two outlines belong to a single object. Here is an alternative way to think about matching: the images of the whole signature functions for γ and γ', denoted $\Sigma = \mathcal{S}(\{\gamma\} \times \mathbb{P}^2)$ and $\Sigma' = \mathcal{S}(\{\gamma'\} \times \mathbb{P}^2)$, form *signature surfaces* in the feature space $\mathcal{F}$. If γ and γ' come from the same object, then the intersection $\Sigma \cap \Sigma'$

yields the consistency hypothesis between the training and test outlines, and its preimage yields the unique baseline joining the camera centers of the training and test images (Figure 1). Thus, a hypothesis of a possible match for recognition is equivalent to a hypothesis of the epipolar geometry of camera pairs in the scene.

So far, we have only discussed matching between a pair of outlines. In principle, since any two views of the same object can be connected by a baseline, it is always possible to match a novel view of an object given a single training image. In practice, however, a single view of an object may be ambiguous, and widely separated pairs of views may fail to produce descriptive features. For these reasons, we should collect training sequences consisting of a few representative views of each object. A recognition algorithm that works with multiple training outlines and a single test outline will have the following conceptual structure:

1. **Training**.
   a) **Feature Extraction.** For each training object i, acquire a training set of outlines $\Gamma_i = \{\gamma_{ij} \mid j = 1, \ldots, m_i\}$ and compute the signatures $\Sigma_{ij} = \mathcal{S}(\{\gamma_{ij}\} \times \mathbb{P}^2)$.
2. **Testing**.
   a) **Feature Extraction.** Acquire a test outline γ' and compute the signature function $\Sigma' = \mathcal{S}(\{\gamma'\} \times \mathbb{P}^2)$.
   b) **Matching**. For each training object i and outline j, compute the intersections $\Sigma' \cap \Sigma_{ij}$. If γ' is a view of object i, then each $\Sigma' \cap \Sigma_{ij}$ should consist of a unique feature vector. Otherwise, each $\Sigma' \cap \Sigma_{ij}$ should be empty.

3 Feature Space

Consider the set of all lines passing through the epipole that are also tangent to the contour. These lines back-project to planes passing through the baseline that are tangent to the object at isolated *frontier points* (Figure 2). The points of epipolar tangency on the image contour are projections of these frontier points, and it is well known that they are the only true stereo matches between a pair of view-dependent contours [8]. Even though a single image does not constrain the depth of frontier points in space, it is still possible to reconstruct the tangent epipolar planes by back-projecting the observed tangent epipolar lines. Notice that the epipolar planes are defined by the baseline and the geometry of the object — they do not depend on image plane orientations or on the exact positions of the camera centers along the baseline. Thus, we can derive feature vectors for baselines by computing the tangent epipolar planes associated with them. The kinds of features we can use — projective, affine, or Euclidean — depend on the imaging model we wish to adopt. In the following subsections, we briefly review these three models in turn. Along the way, refer to Figure 3 for an example of each kind of feature vectors.

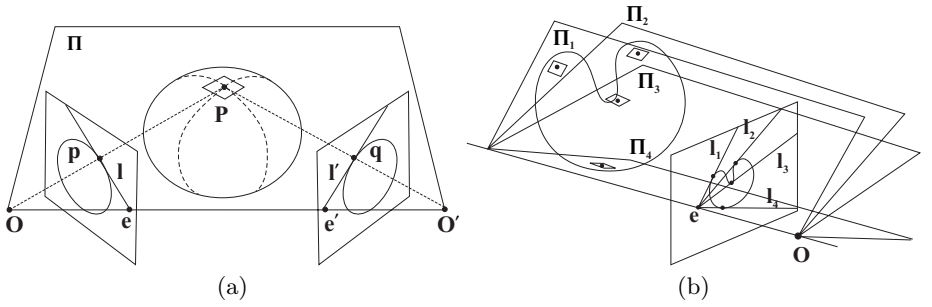


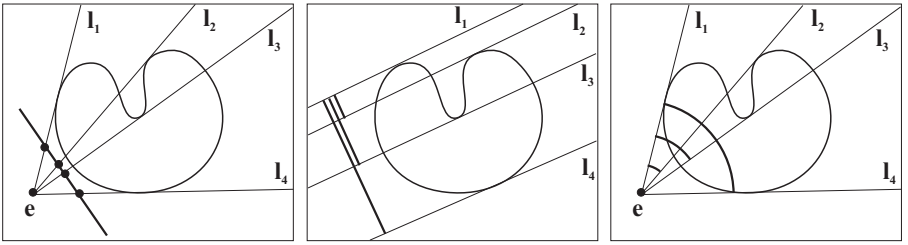
Fig. 2. (a) Epipolar geometry of frontier points. The epipolar plane $\mathbf{\Pi}$ is defined by camera centers $\mathbf{O}$ and $\mathbf{O}'$, and the frontier point $\mathbf{P}$. $\mathbf{\Pi}$ intersects the image planes in two lines $\mathbf{l}$ and $\mathbf{l}'$ that pass through the epipoles and are tangent to the respective contours at the projections $\mathbf{p}$ and $\mathbf{q}$ of the frontier point. (b) Planes $\mathbf{\Pi}_1$, $\mathbf{\Pi}_2$, $\mathbf{\Pi}_3$, and $\mathbf{\Pi}_4$ pass through the baseline and are tangent to the object at four frontier points. These planes intersect the image plane in epipolar tangents $\mathbf{l}_1$, $\mathbf{l}_2$, $\mathbf{l}_3$, and $\mathbf{l}_4$. Note that the epipole $\mathbf{e}$ is hypothetical — it does not correspond to a second camera center.

3.1 Projective Cameras

When the internal camera parameters are unknown, a pinhole camera allows us to measure only the properties of the image that remain invariant under projective transformations. In particular, projective measurements are sufficient to define the epipolar geometry between pairs of cameras. For any two cameras along a fixed baseline, the pencil of epipolar lines tangent to the contour is the projection of the same pencil of planes through the baseline. The cross-ratio of four tangent epipolar planes through this baseline is the same as the cross-ratio of the corresponding epipolar lines observed by any camera along the baseline. Given four or more lines, we can compute all possible cross-ratios between each quadruple of lines and store these cross-ratios in a feature vector.

3.2 Affine Cameras

Affine cameras are cameras whose centers and focal planes are located on the plane at infinity in three-dimensional space [6]. Affine projection preserves parallelism and maps points on the plane at infinity to points on the line at infinity. The notion of epipolar geometry still applies to the affine case: the baseline between two affine cameras is a line at infinity, and since all the epipolar planes intersect in this line, they are actually parallel to each other. In the image, epipolar lines are also parallel, and the epipoles can be thought of as direction vectors. An affine epipole has only one degree of freedom, instead of the two degrees of freedom in the perspective case, and this reduces the intrinsic dimension of the feature space [9]. As for vectors of invariants in the feature space, they are naturally given by ratios of distances between parallel tangent epipolar planes.



(a) Cross-ratio (b) Distance ratios (c) Angles

Fig. 3. An example of different kinds of feature vectors for an epipole with four tangents. (a) Projective (uncalibred cameras): $f = (\text{Cross}(\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3, \mathbf{l}_4))$. (b) Affine: $f = (\text{Dist}(\mathbf{l}_1, \mathbf{l}_2)/\text{Dist}(\mathbf{l}_1, \mathbf{l}_4), \text{Dist}(\mathbf{l}_1, \mathbf{l}_3)/\text{Dist}(\mathbf{l}_1, \mathbf{l}_4))$. (c) Euclidean (calibrated cameras): $f = (\text{Angle}(\mathbf{l}_1, \mathbf{l}_2), \text{Angle}(\mathbf{l}_1, \mathbf{l}_3), \text{Angle}(\mathbf{l}_1, \mathbf{l}_4))$. The angles are not between the lines themselves, but between the corresponding epipolar planes in space. Note that the different feature vectors have dimensions 1, 2, and 3, respectively.

3.3 Euclidean Cameras

Many reliable procedures exist for measuring the internal parameters of the camera (skew, magnification factors, and image center) [6]. Internal calibration gives us a mapping from points in the image plane to lines through the camera center in three-dimensional Euclidean space, expressed in a canonical coordinate system attached to the camera. The projection matrix of the internally calibrated camera may be written as $\mathbf{M} = \mathbf{K}[\mathbf{I} \mid 0]$ where $\mathbf{K}$ is the 3×3 matrix of internal parameters. Then for each line $\mathbf{l}$ tangent to the contour and passing through a particular epipole position we obtain a plane

$$\mathbf{\Pi} = \mathbf{M}^\top \mathbf{l} = \begin{pmatrix} \mathbf{K}^\top \mathbf{l} \\ 0 \end{pmatrix}$$

in the canonical camera system. Given coordinate vectors of two tangent epipolar planes $\mathbf{\Pi}_1 = \mathbf{M}^\top \mathbf{l}_1$ and $\mathbf{\Pi}_2 = \mathbf{M}^\top \mathbf{l}_2$, we may measure their angle as

$$\cos\theta = \frac{\mathbf{l}_1^\top (\mathbf{K}\mathbf{K}^\top) \mathbf{l}_2}{\sqrt{\mathbf{l}_1^\top (\mathbf{K}\mathbf{K}^\top) \mathbf{l}_1} \sqrt{\mathbf{l}_2^\top (\mathbf{K}\mathbf{K}^\top) \mathbf{l}_2}} \ . \tag{1}$$

Given a contour γ and a fixed epipole position $\mathbf{e}$, how do we define the corresponding value of the signature function? Consider the ordered set $(\mathbf{l}_1, \ldots, \mathbf{l}_n)$ of lines that pass through $\mathbf{e}$ and are tangent to the contour (the ordering is circular about $\mathbf{e}$, with $\mathbf{l}_1$ serving as a specially chosen reference line). The planes formed by back-projecting the lines make up a corresponding ordered set, denoted $(\mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_n)$. Let θ_i be the angle in space between $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_{i+1}$, computed according to (1). Finally, we are ready to define the value of the signature function as $\mathcal{S}(\gamma, \mathbf{e}) = (\theta_1, \ldots, \theta_{n-1})$.

In stating the above definition, we have left a few things deliberately unspecified. For one, the order of the angles in the feature vector depends on the

choice of the reference line and on the circular orientation convention (clockwise vs. counterclockwise). In addition, the number n of angles is not a global constant; it may vary for different contours and positions of the epipole. Because of self-occlusion, the number of tangent epipolar planes may actually vary for different camera positions along the same baseline. We will return to these issues in Sections 4.1 and 5.2.

Overall, angles have significant advantages over cross-ratios as primitives making up the feature space. We need fewer measurements to compute angles — only two epipolar tangents suffice, whereas we need at least four to get a cross-ratio. As a result, the "calibrated" feature space has higher dimension than the "uncalibrated" one, which improves the ability to discriminate between different objects at recognition time. For the rest of the paper, we will focus on the calibrated case.

4 Properties of the Signature Function

In the following sections, we briefly describe the smoothness and continuity properties of the signature function, and present informal arguments about the existence and uniqueness of matches in feature space.

4.1 Critical Events

For the rest of this section, let us regard the contour γ as being fixed, so that the signature function depends only on the epipole position $\mathbf{e}$. For a given $\mathbf{e}$, the number of angles in the feature vector $(\theta_1, \ldots, \theta_{n-1})$ is one less than the number of epipolar tangents $(\mathbf{l}_1, \ldots, \mathbf{l}_n)$, and n is also a function of $\mathbf{e}$. For generic epipole positions, a contour will have an even number of epipolar tangents, so the corresponding feature vector will have odd dimension. This dimension will remain constant if we perturb the epipole a little, unless the epipole lies on a *critical event boundary*: the contour itself, an inflectional tangent, or a bitangent to the contour. Crossing an inflectional tangent or the contour itself will make a pair of lines appear or disappear (increasing or decreasing the dimension of $\mathcal{F}$ by two), while crossing a bitangent will reverse the order of a pair of lines (giving no net change in dimension). Away from critical boundaries, however, the values of the angles $(\theta_1, \ldots, \theta_{n-1})$ vary smoothly as a function of the epipole position. Thus, even though the signature surface $\Sigma \subset \mathcal{F}$ may have a very complicated global structure, with different subsets immersed in spaces of a different dimension, its local structure is quite simple. If $\mathbf{e}$ is a generic epipole position giving rise to n epipolar tangents, then inside a sufficiently small neighborhood of $\mathbf{e}$, $\mathcal{S}$ is an immersion of $\mathbb{R}^2$ into $\mathbb{R}^{n-1}$.

4.2 Matching and the Intersection of Signature Surfaces

Let us take two contours γ and γ' and consider the intersection Σ'' of their signature surfaces, $\Sigma = \mathcal{S}(\{\gamma\} \times \mathbb{P}^2)$ and $\Sigma' = \mathcal{S}(\{\gamma'\} \times \mathbb{P}^2)$. Take some $f \in \Sigma''$

where $\mathcal{F}$ is locally m-dimensional (that is, $f \in \mathbb{R}^m$). Then there exist $\mathbf{e}, \mathbf{e}' \in \mathbb{P}^2$ such that $f = \mathcal{S}(\gamma, \mathbf{e}) = S(\gamma', \mathbf{e}')$. Moreover, we can find neighborhoods U and U' of $\mathbf{e}$ and $\mathbf{e}'$, respectively, where $F = \mathcal{S}(\{\gamma\} \times U)$ and $F' = \mathcal{S}(\{\gamma'\} \times U')$ are two-dimensional surfaces in $\mathbb{R}^m$. If we expect F and F' to intersect transversally, then additivity of codimension [5] yields the following:

$$\begin{aligned} m - \dim(F \cap F') &= m - \dim\, F + m - \dim\, F' = 2m - 4 \\ \dim(F \cap F') &= 4 - m\ . \end{aligned}$$

Thus, for $m > 4$, any transversal intersection of two signature surfaces would have to be empty (note that since m can only be odd, we need not be concerned with the case $m = 4$). In other words, if we take two arbitrary contours γ and γ' and a random feature vector f consisting of five or more angle values, we will not find $\mathbf{e}$ and $\mathbf{e}'$ such that $\mathcal{S}(\gamma, \mathbf{e}) = \mathcal{S}(\gamma', \mathbf{e}')$.

Observation 1. If the feature space has sufficiently high dimension, the possibility of "accidental" matches is in principle ruled out.

But what if γ and γ' are outlines of the same object seen from two different viewpoints? Then the baseline connecting these viewpoints gives two true epipole positions $\mathbf{e}$ and $\mathbf{e}'$. Clearly, there exists a unique set of planes that are tangent to the object and pass through this baseline. If we assume the object is transparent, then we will be able to observe exactly the same planes for γ and γ' by looking at the respective epipolar tangents. In this case, we must have $\mathcal{S}(\gamma, \mathbf{e}) = \mathcal{S}(\gamma', \mathbf{e}')$.

Observation 2. For transparent objects, signatures will always match at true epipole positions for two different views of the same object.

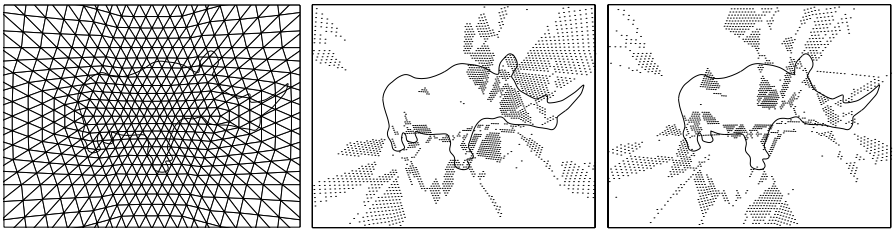
Combined, the two observations above suggest that a match between signature functions for two epipoles in two images indeed offers strong evidence of consistency between two outlines. As long as the dimension of the feature space is high enough, our ability to discriminate approaches the idealization of Section 2. Nevertheless, we cannot claim that a signature match exists *if and only if* γ and γ' are two views of the same object, and $\mathbf{e}$ and $\mathbf{e}'$ are the projections of the camera centers that produced γ' and γ, respectively. Various non-generic properties of the contours, such as symmetries, may conspire to produce multiple signature matches. A far more important problem, however, is self-occlusion — as noted in Section 3.3, two camera positions along the same baseline may fail to see the same epipolar tangents, when some of them become obscured by parts of the surface. In the next sections, we discuss the implementation of our approach, and show how to deal with occlusion.

5 Implementation

5.1 Sampling of Epipoles

We represent signature functions for a small number of input pictures by sampling the two-dimensional space of possible epipoles. We find a set of uniformly

distributed viewing directions by recursively tessellating a unit sphere, and then project these directions onto the image plane (directions lying on the focal plane project to epipoles at infinity). Figure 4 (a) shows a tessellation of the sphere projected onto the image plane. After choosing a sampling pattern of a given



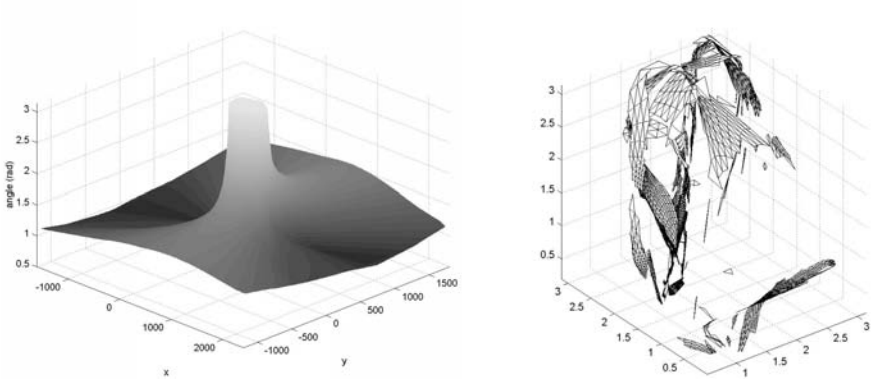
(a) (b) (c)

Fig. 4. (a) A synthetic image of a rhino with a projected 1313-vertex triangulation of the sphere overlayed. (b) Sample points from a 20609-vertex triangulation with 15-dimensional feature vectors. (c) Sample points with 17-dimensional feature vectors.

density, we find all epipolar tangents and compute the signature functions for each sample point. During a pre-processing step, contours are segmented using thresholding followed by Gaussian smoothing [9]. To facilitate the computation of epipolar tangents, the contours are represented as cubic B-splines. Recall from Section 4.1 that for different epipole positions, the number of epipolar tangents and the dimension of feature vectors vary as certain critical boundaries are crossed. Patterns of sample points with feature vectors of different dimension shown in Figures 4 (b) and (c) reveal these boundaries. To visualize the computed signature functions, refer to Figure 5: part (a) shows a plot of the largest angle in the feature vector as a function of the epipole, and part (b) shows some patches of a 15-dimensional signature surface projected into three dimensions.

5.2 Ordering of Feature Vectors

Let us return to the definition of a feature vector given in Section 3.3. At that stage, we have not committed to a choice of a reference plane $\mathbf{\Pi}_1$ or to the orientation of the circular ordering of planes around the baseline. However, for best recognition results, the choice of $\mathbf{\Pi}_1$ should not be arbitrary. Whenever the baseline passes outside the convex hull of the object, we can identify two *extremal planes* that make contact with the object only at the respective frontier points. These planes are robust to self-occlusion in opaque objects — they will necessarily be observed between any two views on the same baseline. By contrast, frontier points due to non-extremal planes can become occluded (in addition, segmentation algorithms usually miss parts of the contour that are visible, but



(a) (b)

Fig. 5. (a) Angle (in radians) between extremal planes as a function of epipole position (in pixel coordinates) for the rhino image shown in Figure 4. Note that the extremal angle approaches π as the epipole approaches the contour. (b) A three-dimensional immersion of a 15-dimensional subset of the signature surface for the rhino contour.

fall inside the silhouette). For these reasons, our implementation arbitrarily selects one of the two extremal planes as the reference, and computes the angles $(\theta_1, \ldots, \theta_{n-1})$ with respect to this reference plane. While matching feature vectors from two images, it is impossible to determine whether the two reference planes correspond to each other in space, or whether the reference plane in one image corresponds to the second extremal plane in the other image. For each of the two possible orderings, we compute a matching cost as described in the next section, and select the smaller cost as the "winner".

When the baseline enters the convex hull of the object, there are no extremal planes. In this case, matching becomes more combinatorially complex, and more difficult to implement. However, since only a small fraction of all sample epipoles fails to produce extremal planes, excluding these points from matching has a negligible effect on performance.

5.3 Matching Feature Vectors

In the idealized recognition framework of Section 2, matching reduces to finding intersections of signature surfaces. Unfortunately, this formulation is difficult to implement directly. Since we are using a sampled representation of signature surfaces, we cannot locate exact matches by simply comparing discrete feature vectors. Also, Observation 2 of Section 4.2 is not true for opaque objects. Self-occlusion can make some tangent planes invisible from a particular camera position along the baseline, and introduce T-junctions that show up as false frontier points on the silhouette. Because of these effects, a successful matching algorithm must be able to compare feature vectors with different numbers of components and find subsequences of these vectors that minimize some matching cost. To

this end, we have implemented a dynamic programming algorithm that, given two feature vectors of length m and n, $f = (\theta_1, \ldots, \theta_m)$ and $f' = (\theta'_1, \ldots, \theta'_n)$, finds two subsequences of length k, $\tilde{f} = (\theta_{i_1}, \ldots, \theta_{i_k})$ and $\tilde{f}' = (\theta'_{j_1}, \ldots, \theta'_{j_k})$ that minimize the average distance function

$$\mathrm{D}(\tilde{f}, \tilde{f}') = \frac{1}{k} \sum_{l=1}^{k} |\theta_{i_l} - \theta'_{j_l}| \ . \tag{2}$$

The subsequence length k can either be given to the matching algorithm as a parameter, or used as another optimization variable. The dynamic programming formulation is relatively efficient, and it is the natural way to enforce ordering constraints — e.g., the extremal angles always have to match, and the indices in the two subsequences must increase monotonically.

5.4 Matching Signature Surfaces

Once the matching score for a pair of feature vectors has been defined, we can proceed to match pairs of signature surfaces. In Section 4.2, we established that, provided the dimension of the feature space is sufficiently high, we can expect a unique match between two signature surfaces $\Sigma = \mathcal{S}(\{\gamma\} \times \mathbb{P}^2)$ and $\Sigma' = \mathcal{S}(\{\gamma'\} \times \mathbb{P}^2)$. Thus, in the implementation, it is sufficient to look for a single pair of "closest" feature vectors. The signature matching cost is then simply

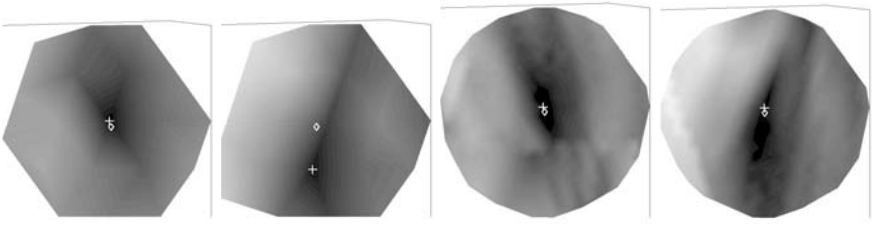
$$\mathrm{C}(\Sigma, \Sigma') = \min_{f \in \Sigma, f' \in \Sigma'} \mathrm{D}(f, f') \ .$$

In practice, because of measurement noise and discretization error due to sampling, C will not vanish even if Σ and Σ' intersect. When comparing a test signature surface Σ' to training surfaces $\Sigma_1, \ldots, \Sigma_m$, we can assign matches based on minimum cost:

$$\mathrm{Match}(\Sigma') = \arg\min_{\Sigma_i} \mathrm{C}(\Sigma_i, \Sigma') \ . \tag{3}$$

Let $f = \mathcal{S}(\gamma, \mathbf{e})$ and $f' = \mathcal{S}(\gamma', \mathbf{e}')$ be two feature vectors giving the lowest-cost match. The two points $\mathbf{e}$ and $\mathbf{e}'$ represent a hypothesis of the epipolar geometry between the views that produced outlines γ and γ'. When full calibration is available, comparing the locations of these points to the true epipole positions allows us to verify the matching procedure. To conduct the verification, we experimented with a synthetic rhino data set (Figure 4 shows one of the rhino images). First, we computed matching costs for the signature of the true epipole in one view and the signatures of all sample points in a second view. Figure 6 shows the resulting plots for two different sampling resolutions. Well-defined local minima exist in the vicinity of the true epipoles, although the discrepancies between the minima and the true matches vary with the quality of the sampling.

Next, we computed cost functions for all pairs of sample points whose viewing directions fall within 10° of the true epipoles. The results are summarized



View 1, 4° sampling View 2, 4° sampling View 1, 1° sampling View 2, 1° sampling

Fig. 6. Matching costs between true epipoles and sample points plotted on the sphere for directions within 10° of the true match. Darker shading indicates lower cost. The local minimum of the sampled cost function is marked with a cross, and the true epipole location is marked with a diamond.

in Table 1. Our experiments show that the minimum cost over all pairs of sampled feature vectors may be an order of magnitude larger than the cost for the true match (of course, the actual numerical values are an artifact of our definition of the cost function). However, as sampling density increases, the minimum cost computed over all pairs of sample points approaches the global minimum (compare rows 1 and 2 of the table). By examining row 3, we can also see that denser sampling improves the accuracy of hypothesized epipoles. Interestingly, though, the minimum cost match is not found at sample points that are closest to the true epipole directions. Overall, our results confirm that it is in principle possible to find reliable epipole estimates through matching signature surfaces — empirically, the cost of the true match always appears to be the global minimum. However, the actual quality of local minima found by our algorithm is dependent on the density of the sampling.

Table 1. The effect of sampling density on local minima of the matching cost function. The third row shows the angle differences between true epipoles and minimum-cost sample points in views 1 and 2, respectively.

	Sampling density 4° (1313 points)	Sampling density 2° (5185 points)	Sampling density 1° (20509 points)
Actual min. cost ($\times 10^4$)	4.136	4.136	4.136
Sampled min. cost ($\times 10^4$)	20.078	12.147	4.803
Angle discrepancy	7.34° and 7.57°	5.41° and 4.99°	3.91° and 3.68°

5.5 Recognition

In this section, we present a matching experiment on a real data set containing two views each of three objects: a toy dinosaur, a gargoyle statuette, and a

cowboy (see Figure 7). The data set shows a substantial amount of self-occlusion: notice, for instance, the tail and the forelegs of the dinosaur, and the left ear and wing of the gargoyle. For each input picture, signature surfaces were computed at 4° resolution. As Table 1 indicates, the local minima of the cost function computed at this sampling density are not very reliable. To diffuse the sampling artifacts and to pool evidence from multiple locations in the cost landscape, we modified the matching criterion of Equation 3 to take the average of a fixed number of the lowest-cost feature matches. Specifically, to classify each outline, we computed the mean of the lowest 50 matching costs of its signature surface with the signature surfaces of every other outline, and picked the smallest-cost outline as the winner. Figure 8 presents the complete matching statistics. Note that each outline is correctly assigned to the other outline from the same object.

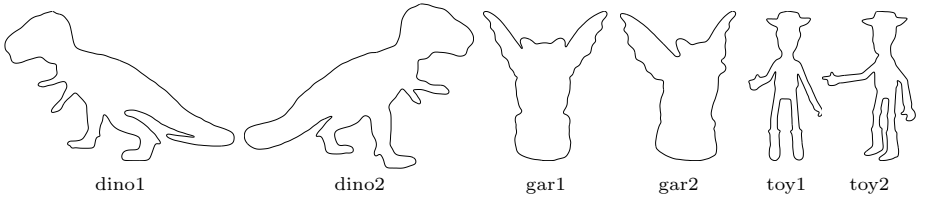


Fig. 7. Outlines of three objects used in the recognition experiment.

Our recognition experiment allows us to draw several conclusions. First of all, dense sampling does not appear to be necessary for successful matching. Even though the lowest-cost matches may be far away from the true epipoles, the relative magnitudes of the costs give a good indication of proximity between different signature surfaces. Secondly, reliability of matching depends on the complexity of the contours. For instance, the toy outlines are the most complex, giving rise to the highest-dimensional signature surfaces. Feature vectors from these surfaces offer a large number of possible combinations for matching, raising the likelihood that a spurious low-cost match will be found.

6 Discussion and Conclusions

The preliminary implementation of Section 5 confirms the validity of our recognition framework, but it cannot serve as a prototype for a working real-world system. To make our method truly practical, we need to address several issues.

– **Segmentation**: since contour extraction is not the focus of the current paper, we assume that all input images can be segmented using naive techniques like thresholding. This restrictive assumption is not unique to our approach, but is common to most silhouette-based recognition or reconstruction schemes. Overall, the development of robust and general segmentation algorithms remains a significant challenge.

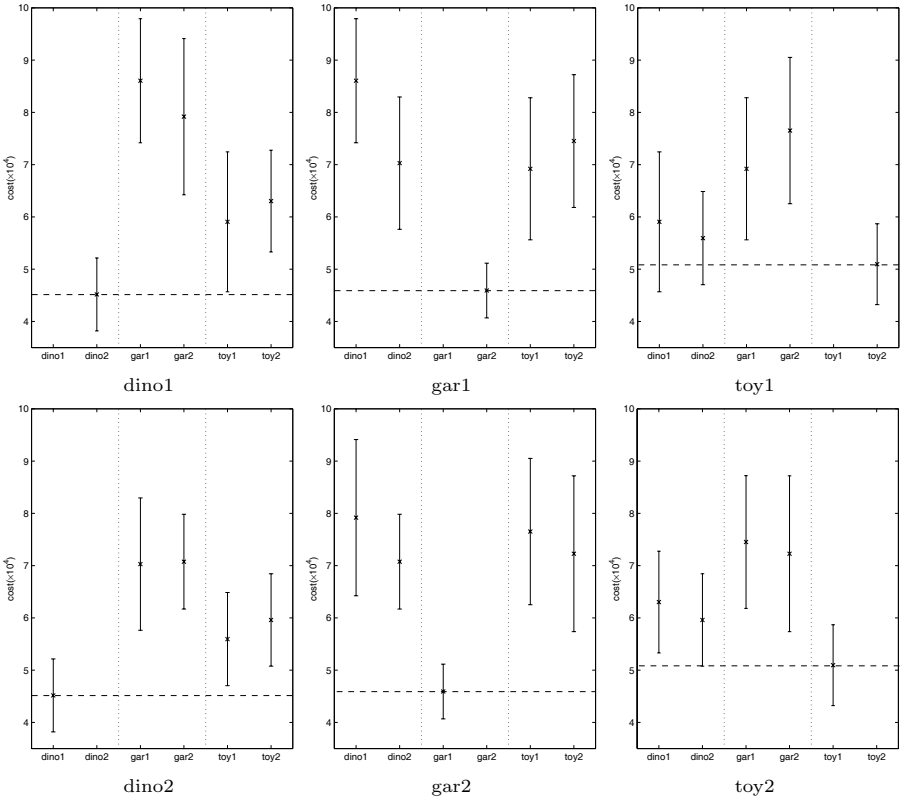


Fig. 8. Mean and standard deviation of matching costs for each test outline vs. all the other test outlines. The dashed horizontal lines indicate the lowest cost matches.

- **Occlusion and clutter**: the feature matching algorithm of Section 5.3 only deals with measurement noise and self-occlusion. We are currently modifying our framework to account for occlusion of the target object by other objects, and for segmentation errors due to background clutter.
- **Efficiency**: our implementation involves sampling two-dimensional sets of epipoles, and matching between all pairs of feature vectors in two images. We need to optimize these computationally expensive tasks, or develop alternative signature function representations and matching procedures.

One interesting extension of our approach is to combine the discrete feature matching procedure with nonlinear optimization methods that solve for camera motion based on frontier points [4]. The main problem with these methods is initialization — it is difficult to find an initial guess of epipole positions that would make the system converge to the right solution. We could start an optimization at the local minima produced by our matching algorithm, and use an iterative technique to improve the estimates of the epipoles.

Another important long-term direction is class-based object recognition. In this paper, we developed a representation framework that captures the geometric constraints between different views of *a single object instance*. A far more challenging question is, what geometric features derived from image data would allow us to classify pictures drawn from *an object category*? Developing algorithms that reason directly about 3D geometry, rather than about 2D image patterns, may be the key to building recognition systems that discriminate between classes of objects related by similarity of 3D shape.

Acknowledgments. This work was supported in part by the Beckman Institute, the National Science Foundation under grants IRI-990709 and IIS 00-85980, and a UIUC-CNRS collaboration agreement. We would also like to thank Steve Sullivan for providing the data used in the experiment of Section 5.5.

References

1. R. Basri and S. Ullman, "The Alignment of Objects with Smooth Surfaces", *Proc. of Int. Conf. on Computer Vision*, 1988, pp. 482-488.
2. S. Belongie, J. Malik and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts", to appear, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3), 2002.
3. S. Carlsson, "Order Structure, Correspondence, and Shape Based Categories", *Proc. of Int. Workshop on Shape, Contour and Grouping*, 1999, pp. 58-71.
4. R. Cipolla, K. Astrom and P.J. Giblin, "Motion from the Frontier of Curved Surfaces", *Proc. of IEEE Int. Conf. on Computer Vision*, 1995, pp. 269-275.
5. V. Guillemin and A. Pollack, *Differential Topology*, Prentice Hall, Englewood Cliffs, New Jersey, 1974.
6. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, 2000.
7. D. Jacobs, P. Belhumeur and I. Jermyn, "Judging Whether Multiple Silhouettes Can Come from the Same Object", to appear, *Proc. of the 4th Int. Workshop on Visual Form*, 2001.
8. J. Porrill and S. Pollard, "Curve Matching and Stereo Calibration", *Image and Vision Computing*, 9(1), 1991, pp. 45-50.
9. D. Renaudie, D. Kriegman and J. Ponce, "Duals, Invariants, and the Recognition of Smooth Objects from their Occluding Contour", *Proc. of Eur. Conf. on Computer Vision*, 2000, pp. 784-798.
10. C. Schmid, "Constructing models for content-based image retrieval", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
11. H. Schneiderman and T. Kanade, "A Statistical Method of 3D Object Detection Applied to Faces and Cars", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition* 2000, pp. 746-751.
12. M. Weber, M. Welling and P. Perona, "Towards Automatic Discovery of Object Categories", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2000.