Efficient Encoding of Local Shape: Features for 3-d Object Recognition

Paul G. Gottschalk Trevor Mudge

Robotics Research Laboratory, EECS Department, University of Michigan, Ann Arbor, MI, 48109.

Abstract

To date, algorithms designed to recognize 3-d objects from intensity images have either employed global features or local features based on straight line segments. Global features limit recognition to objects that are completely visible and segmented from the background; these are often unrealistic conditions in machine vision domains. Local features based on line segments are somewhat more flexible since partially visible objects may be recognized, however, the requirement that objects will have sufficient straight lines to ensure robust recognition is also limiting. In this paper, we discuss a set of features, called *scale-invariant critical point neighborhoods*, or SICPN's, which are more generally applicable to 3-d object recognition. SICPN's efficiently encode the local shape of edge segments near points of high curvature. To within variations caused by image noise and discretization, SICPN's are invariant to image plane translations, rotations, and scaling. These invariant properties enhance the utility of these features in 3-d recognition algorithms. Furthermore, we show that SICPN's have many other desirable characteristics including informativeness, ease of detection, and compact representation. In addition, we empirically demonstrate that SICPN's are insensitive to noise. The above characteristics are essential if a feature is to be useful for 3-d object recognition.

1 Introduction

Recognition of 3-d objects in 2-d intensity images is an important problem in computer vision which has a large number of practical applications. In order to recognize objects, properties of the objects that allow one object to be distinguished from another, as well as from the background, must be identified. In some domains, such as automatic interpretation of aerial photographs, spectral and textural characteristics are most important. In many other domains, particularly those where the objects to be recognized are man-made, shape is the best distinguishing characteristic. In this paper we describe a class of features called *scale-invariant critical point neighborhoods*, or SICPN's, that efficiently encode the shape of edge contours.

2 Requirements of Features for 3-d Object Recognition

The process of 3-d object recognition from intensity images can be viewed as a search through the space of all possible instantiations of object models for those instantiations which, when graphically rendered, best agree with the image data. The size of this space is immense. For example, considering only rigid objects (i.e., not articulated or deformable objects), and ignoring the degrees of freedom inherent in the lighting of the scene and in the camera, there are seven axes of variation that can change how the object appears in an image. The first is the discrete axis of model identity, i.e., whether the object is a plane, car, tin can, etc. The next six constitute the continuous space of viewing parameters, and describe the model's position and orientation in 3-d space, which, in turn, affect the image intensity function. It is clear that any search of this space that is not highly efficient will be unlikely to succeed.

Many practical object recognition algorithms can be divided into two types of processes. The first process generates model-transform hypotheses based on the image data. A model-transform hypothesis is a conjecture as to the identity of the model and the viewing parameters that instantiate it in the scene. The second process gathers positive and negative evidence which is used to make a decision about whether the hypothesis is consistent or inconsistent with the image. The simplest form of the second process takes the form of *hypothesis verification* where a detailed comparison of the observed image data and the rendered model instance is performed in the image domain in order to *verify* that the instance of the model explains what has been observed in the image. This leads to the commonly used *hypothesize and verify* paradigm for recognition.

Practical object recognition algorithms employ appropriate *features* to improve their efficiency. A feature is an entity with an associated vector of property attributes. For example, a line segment has associated with it a location attribute, an orientation attribute, and a length attribute. In order to satisfy our goals, good features for shape-based 3-d object recognition must be spatially local or consist of configurations spatially local subparts. There are two reasons for this:

1. Noise and encroachment of the background may cause parts of the object to be distorted or to be missing in the image. Spatially local features will have a smaller chance of being distorted since a small feature stands a good chance of being on a portion of the object that is undistorted.

2. We have the goal of recognizing objects that may be partially obscured by others. As in 1, spatially local features minimize the possibility that the feature will be missing or distorted due to occlusion by another object.

The features that we seek should possess a number of properties in addition to spatial locality They should should:

- be noise resistant
- be informative
- be insensitive or invariant to unimportant viewing parameters
- be easily segmented
- have a compact representation
- minimize the number of image-feature to model-feature correspondences that must be made during the recognition process.

The importance of the property of noise resistance is obvious. The other properties, however, require further explanation.

A feature is informative if its presence in the image data allows a large part of the space of possible object model instantiations to be eliminated from consideration. For example, a single point (such as an edge point) is as spatially local as possible, nevertheless it is an uninformative feature because its presence in the image (by itself) gives no clue as to which object's presence in the scene may have generated it, and only eliminates two degrees of freedom in the space of possible viewing parameters. On the other hand, a pair of directed points (a point with an associated direction is a directed point, such as an edge point with the intensity gradient at that point) is a much more informative feature since such a feature could only be generated by the (far fewer) object model instantiations that can place edge points at the right locations *and* with the proper angles between the gradient vectors and the vector connecting the two points. In practice, even pointlike features require that a neighborhood of the image array be examined to detect them, and therefore are not truly pointlike. In general, there is a tradeoff between the degree of spatial locality that a feature exhibits, and its information content. Given a vision system's constraints on the spatial locality of the features (which, in turn, is driven primarily by requirements on the system's immunity to missing and distorted features), good features will provide the most information possible.

The property of insensitivity or invariance to unimportant viewing parameters is closely related to informativeness. Vision researchers have sought features that are invariant to variations in various viewing parameters such as image plane rotation, translation, and scaling, among others. Insensitivity to variations in lighting conditions is another sought-after property of features. Invariance and insensitivity of features is most desirable in the hypothesis generation phase of of the recognition process. The reason for this is best illustrated by example. Return to shape-based recognition of rigid 3-d objects from single intensity images. In this domain, an ideal (and non-existent) feature would be one that gives a local description of the shape of the surface of the object (in 3-d space) based only upon a portion of the data in the image array. If certain assumptions ar made it is possible to do this, as demonstrated by the work on shape from shading, texture, shadows, etc. Since objects in this domain are distinguished entirely by their shape, such features would allow a recognition system to immediately retrieve (using an appropriate indexing scheme) all objects that have parts of their surfaces with the shape indicated by the image data. Such a feature would allow six degrees of freedom to be eliminated in the search for possible hypotheses, reducing the process to searching the space of object identities alone. In practice, without making overly restrictive assumptions about the nature of the imaging domain, such ideal features are not available when the image sensor measures intensity. Range sensors, on the other hand, can provide such information directly. This is why range sensors are gaining much attention for use in object recognition: they simplify the matching process considerably. The relationship between informativeness and invariance can now be seen clearly. An invariant feature drastically cuts the amount of search necessary for an important part of the recognition process, and hence tends to be more informative than non-invariant features.

Ease of segmentation has often been ignored in choosing features in the past, yet is an important property. No matter how informative a feature is, if it cannot be not reliably detected from the image, then it is not useful for recognition. Perhaps the best example of such features are the perfect junction classes of blocksworld and its derivatives. Such features

allow powerful matching algorithms to be employed to solve difficult problems. However, in practice, it is not known how to reliably detect such features in a real image. Another example is the early approach of using global features. Such methods compute a global feature from an object that is assumed to be totally visible and in some way segmented from the background. The problem of segmentation of the objects from the background has been assumed to have been solved, leaving only the problem of identifying the objects. In practical domains, this assumption is usually far too restrictive. Therefore, the ease of detecting, or segmenting, features is an important consideration.

A compact representation means that the feature has only the information that is important for recognition, with no redundant or useless information. An example of a non-compact representation is enumerating the x-y position of ten sample points on a line as opposed to the much more compact representation of just listing the endpoints. Compactness of the feature's representation ensures that no more data than is necessary is used to store the feature, as well as reducing the amount of data that must be processed by the system.

The final desirable property of good features, that of minimizing the number of image-feature to model-feature correspondence hypotheses, is closely tied to the matching process. Generally, all object recognition methods must perform some kind of explicit or implicit hypothesis generation, during their execution. Generating hypotheses is a *data driven* or *bottom-up* process. When hypotheses are first being generated, there are no hypotheses, and therefore no model-directed expectations about what features are likely to be present in the image. In the absence of such expectations, all image features must be considered equally; none can be rejected as spurious. Under these conditions features that consist of a configuration of spatially local subparts may cause a combinatoric explosion in the hypothesis generation portion of the matching process. To illustrate, return to the example of a feature consisting of a pair of directed points. Assume that the points are detected in some manner independent of each other. This is reasonable since, again, there are no top-down expectations from models at this point. There may be model-independent heuristics, but we will not consider these in this simple example. Then, all possible pairs of these directed edge points must be considered as potentially valid features, though, in fact, many of them will be invalid since they are composed of primitives that belong to distinct objects. If there are N directed edge points detected, then N^2 feature pairs will be generated. In a slightly more general situation where each composite feature consists of k spatially local primitives of a single type of feature detected, then there will be N^k possible composite features. The number of composite features that must be considered by the hypothesis generation process explodes combinatorically if k is too large. There are two ways that the complexity can be reduced without reducing the informativeness of the compound features:

1. Minimize the number of primitive components by making the primitives as informative as possible.
2. Choose primitive components for which there are model-independent guidelines for determining which primitive features are more likely to belong to the same object.

In the previous paragraphs, we have discussed what properties should be possessed by a good feature for 3-d object recognition. In the remainder of the paper, we present a new type of feature we call *scale-invariant critical point neighborhoods*. We will also show that these feature have many of the desirable properties we have just enumerated.

3 Features Employed Previously for 3-d Object Recognition

Features that have been employed in previous systems whose goal is to recognize partially visible 3-d objects based on their shape have been compound features composed of primitives that are line segments, or pointlike features. Lowe [12] used clusters of line segments called *perceptual groupings*. Perceptual groupings were formed based on a model-independent argument of the likelihood that the segments belonged to a single object. ACRONYM [2] was also based on linear primitives, although the system worked only with compound configurations of line segments that formed trapezoidal, hexagonal, and ellipsoidal groupings. Lamdan and Wolfson [11] describe a recognition system that is also based on sets of line segment features, as do Thompson *et al.* [14,4], Horaud [8], and Goad [6].

We argue that, by themselves, line segment features are not good features for the recognition of general 3-d objects. We have several reasons for this belief:

- Objects consisting of many curved surfaces will not have many linear features.
- Line segments are relatively uninformative features. Typically, therefore, compound features must be comprised of several line segments in order to be informative enough for the application. This, in turn, increases the combinatorics

of matching.

- Polygonal approximations of curves are not stable.

Pointlike features have the advantage that they are very local (although a neighborhood of pixels must still have been examined in the image in order to detect them). However, pointlike features are even more uninformative than line segments, having only the attributes of location. A small improvement on the simple point is the directed point, which consists of a point and an associated direction. Huttenlocher *et al.* [9] use compound features consisting of both points and directed points. These points are inflection points (points of zero curvature) in the edge contours. Chien and Aggarwal [3] use configurations of high curvature points, similar to the critical point that form the centers of our SICPN features. However they do not use the local shape as we do. Lamdan *et al.* [10] use lines and points to recognize objects. The points are of two types, one type, which they call "interest points" are points of high or infinite curvature, and so are also related to critical points. The second type are the constituents of a triplet that marks the entrance and deepest point of a concavity on the curve. The methods we mentioned above, are concerned with the recognition of 3-d objects in intensity images. In addition, there many methods concerned with recognition of 2-d objects from intensity images that use pointlike features, however we will not concern ourselves with them here.

Our philosophy is that the primitive elements of features should be as descriptive as possible. The reasons for this are twofold. First, as discussed previously, descriptiveness, if exploited, will tend to reduce the combinatorics of the recognition process. Second, with spatially local primitives, the property of continuity can be exploited. That is, if a feature is detected from a local neighborhood in the image, then it is likely that the feature belongs to a single object. Such a guarantee cannot be made as strongly of the elements of a compound feature. Furthermore, descriptive features make it possible to reduce the combinatorics of recognition without making the set of features too sparse, thus making the recognition process more resistant to occlusion, missing features, and spurious features. Curiously, several 2-d recognition methods have employed features that are composed of highly informative spatially local, shape-based primitive features [7,1], but no 3-d methods that we know of have done so.

4 Scale Invariant Critical Point Neighborhoods

In this section we describe new features for 3-d object recognition called *scale invariant critical point neighborhoods* (SICPN's). SICPN's encode the local shape about *critical points*. While there are several interpretations in the literature of the term critical points, we will use the common interpretation as points of locally maximum or minimum curvature on the edge contours of objects. SICPN's allow complete control over its informativeness by varying the amount of the boundary that is used to compute the feature; indeed, the tradeoff between informativeness and spatial locality is explicit with SICPN's. The ability to control the SICPN's informativeness makes them good primitives for compound features.

A SICPN is computed in a sequence of six steps:

1. Find connected edge contours.
2. Compute the slope-angle versus arclength representation of the edge contours.
3. Detect critical points from the θ-s representation of the contours.
4. Estimate the curvature of the contour at the critical point.
5. Sample the θ-s contour at M equally spaced points in an interval of arclength with the critical point as its center. The sample values are shifted in θ so that the value of the center sample (where the critical point is located) is zero. The length of the interval is chosen to be inversely proportional to the curvature estimate, resulting in a vector of samples that, ideally, is invariant to rotation, translation, and scaling in the image plane.
6. Project the vectors obtained above onto a set of orthogonal vectors that perform the function of filtering noise and data compression.

We now describe each of the six steps in greater detail.

The first step, finding linked edge contours, is straight forward. Any edge detector can be used to find an edge map, followed by a linking process that groups edge elements into sets of contiguous sets, or contours. We use an implementation

of Canny's edge detector that produces, in addition to the edge map, an intensity gradient map. Our linker searches for a contiguous pixel first in the direction indicated by the gradient, and then searches pixels progressively further from the gradient direction. If a contiguous pixel is found, the process is repeated at the new pixel. If none are found, the process stops. We allow only two-connected contours. The linker stores both the x-y location of each edge element in the contour, as well as the value of the gradient there.

After the edge contours in the image are found, we compute their slope-angle versus arclength representation, which we will call the θ versus s or θ-s representation. The ordinate, or θ, in the θ-s representation is the slope angle of the tangent at the point of interest on the contour, measured relative to the horizontal. Similarly,the abscissa, or s, is the arclength measured from some arbitrary point of reference on the contour to the point of interest. The contour is resampled at uniform arclength intervals. The values of θ and the x-y coordinates at the resampling points are computed by linearly interpolating between the nearest points on the contour.

The next step is the detection of the critical points on the contours. As mentioned earlier, critical points are points of maximum absolute curvature on the edge contours of objects. From calculus, the definition of curvature is simply the derivative of the θ-s curve with respect to arclength. In practice, differentiating the θ-s curve and searching for locally maximal absolute values is not robust since the derivative operator amplifies high frequency noise. Instead, we use a multiscale version of a 1-d derivative of a Gaussian edge detector to detect the critical points. This scheme is related to the 2-d edge detector described by Schunck [13]. The use of multiple scales has two major advantages:

- The critical points are stable over a wider range of scale than with a single scale detector.
- With a single scale detector, the localization of the critical point degrades as the σ of the Gaussian increases, while the signal to noise ratio degrades with decreasing σ. With our multiscale method, the localization of the edge detector is nearly as good as the smallest scale, but has the noise immunity of the largest.

A major source of error in the following steps is the error in the localization of the critical points. This is why we chose to use the multiscale critical point detector.

Our multiscale critical point detector detects critical points in the following manner. First, the original θ-s curve is convolved with a progression of l Gaussian derivatives, each having a σ_i such that $\sigma_i = \rho\sigma_{i-1}$, where ρ is the ratio between two scales. The resulting set of curves, $\alpha_i(s), i = 0...l$, have maxima where the rate of change of the slope angle is large at the scale of the particular Gaussian derivative that was used to obtain it. The l curves are then multiplied point by point to yield a curve $\beta(s) = \prod_{i=1}^{l} \alpha_i(s)$ whose local maxima we will define as critical points. Peaks that appear in a few of the $\alpha_i(s)$ will be amplified in $\beta(s)$ with respect to noise, since it is unlikely that noise peaks will appear in more than one of the $\alpha_i(s)$. In addition, the smaller scale curves will define the width of the peaks in the $\beta(s)$ curve, providing the best localization of the critical points. For more details on the advantages of this method, see Schunck [13]. As an illustration of the above method, Fig. 1 shows the result of applying the multiscale critical point detector to an actual object, a baby's rattle. The results of using three scales and a single scale are shown. The advantages are most apparent in the nature of the $\beta(s)$ curve. There is much more noise in the single scale detector.

As mentioned previously, one of the properties of a good feature is insensitivity or invariance to changes in the viewing parameters. Since the θ-s contours are derived from the results of detecting edges, they already have a degree of insensitivity to changes in lighting. An additional virtue of the θ-s representation of edge contours is that it can be made invariant to rotations and translations of the image plane. To see this, first assume that there exists a reference point on the contour that can be reliably detected, a critical point for example, from which the arclength measurements will be made. Since all measurements are relative to the point of reference, translation invariance is automatic. As a contour is rotated in the image plane, the ordinate, θ, of the θ-s representation experiences a shift that is proportional to the rotation. More precisely, let $\theta(s)$ be the θ-s representation that has been parameterized by the arclength from the reference point (at $s = 0$). If the contour is rotated by γ, then the new contour will have the θ-s representation $\theta'(s) = \theta(s) + \gamma$. We can normalize the curve with respect to the rotation by subtracting the value of theta at the reference point from $\theta'(s)$, i.e., $\theta'(s) - \theta'(0)$. If the contour is rotated now, the result is $(\theta(s) + \gamma) - (\theta(0) + \gamma)$ which is identical to the original. Thus, assuming a reference point can be located on a θ-s curve, the representation can be normalized so that it is invariant to image plane rotation and translation.

The above discussion indicates that it is possible to compute a shape-based local feature that is invariant to image plane rotations and translations. This can be done in the following manner. Given a contour and a critical point that has been detected on it, as before parameterize the θ-s representation of the contour so that the critical point is at the point where

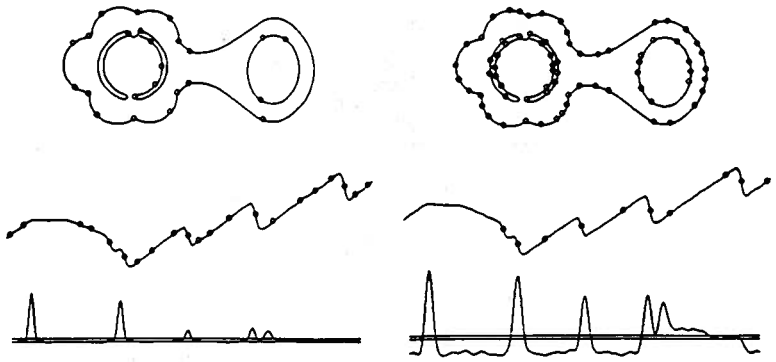


Figure 1: Shown above is the result of applying our multiscale critical point detector to an image of a baby's rattle. On the left is the sequence of curves that results from using three scales, with $\sigma = 4$, 6 and 10 samples respectively. On the right is the result of applying a single scale detector to the same edge contour. The sequnce of curves, from top to bottom, are: the x-y representation of the contour with detected critical points shown; $\theta(s)$ with detected critical points shown; $\beta(s)$. Note the considerably larger amount of noise present in the single-scale $\beta(s)$ curve which results in many spurious critical points being detected.

$s = 0$. Normalize the contours as indicated in the previous paragraph, using the critical point as the reference point. For simplicity of notation, we will hereafter assume that $\theta(s)$ refers to a representation that has been normalized with respect to rotation so that $\theta(0) = 0$. Now, sample the at a fixed number of samples, say M, over the interval $[-\delta, \delta]$ to yield an M dimensional vector. Since $\theta(s)$ is invariant with respect to rotation and translation, the vector will be as well. Thus, the local shape of edge contours about critical points can be compared directly using some metric over $\Re^M$.

Invariance to image plane rotation and translation is certainly helpful, but it is also possible compute a feature which characterizes the local shape of a contour and that is invariant to scale changes as well. As was previously argued, features that possess additional invariant properties help to reduce the size of the search space during the recognition process. An object is not precisely scaled when it is translated to a point nearer or farther from the camera. In reality, perspective projection introduces distortions in the shape. However, if the distance to the object is large in comparison to its depth, then the perspective projection can be approximated by a parallel projection followed by a scaling operation. We will assume that this is a reasonable approximation in any system that will employ SICPN's as features.

Features that are invariant to scale can be constructed as a variation on the theme of the vector feature described two paragraphs previously. To see how to accomplish this, let $\theta(s)$ be the θ-s representation of the original contour, parameterized such that a critical point is at $s = 0$, normalizing it with respect to rotation as described above. Similarly, let $\theta_a(s)$ be the θ-s representation of the contour after scaling the contour by a factor of a. Then,

$$\theta_a(as) = \theta(s). \tag{1}$$

The final SICPN feature is obtained by sampling the θ-s representation at M equally spaced points in an interval about the critical point at $s = 0$. In order to normalize out the effects of scaling the contour, we must scale interval proportionately, i.e., if the interval $[-\delta, \delta]$ is used to sample $\theta(s)$, then the interval that should be used for $\theta_a(s)$ is $[-a\delta, a\delta]$. In order to demonstrate the invariance to scale, let s_i, $i = 1...M$ be the sample points on the unscaled contour and therefore the feature vector will have the components $\theta(s_i)$, $i = 1...M$. The points on the scaled contour are then as_i, $i = 1...M$, implying that the scaled feature vector will have components $\theta_a(as_i)$, $i = 1...M$. Using Eq. 1 yields $\theta_a(as_i) = \theta(s_i)$, showing that the components of the scaled and unscaled feature vectors will be identical.

The rotation, translation and scale-invariant feature vector that we have described above is not yet a true SICPN, as there remain further computations to perform on it. In the remainder of the paper, we will refer to these feature vectors as *shape vectors* since they describe the shape of the neighborhood of a contour surrounding a critical point.

We have shown, in theory, how to compute the shape vectors upon which the computation of SICPN's themselves rests. However, in practice, there remains the problem of how to *reliably* obtain the size of the sampling interval. We do this by estimating the slope of the $\theta(s)$ curve at $s = 0$. To estimate the slope of the curve, a line must be fitted to $\theta(s)$ in the neighborhood of $s = 0$ where the curve is approximately linear. A simple least-squares fit using a fixed number of sample points to compute the fit is not suitable because, at some scales, the sample points may include portions of the curve that are not approximately linear, and therefore the fit will cease to provide a good estimate of the slope at the critical point. We

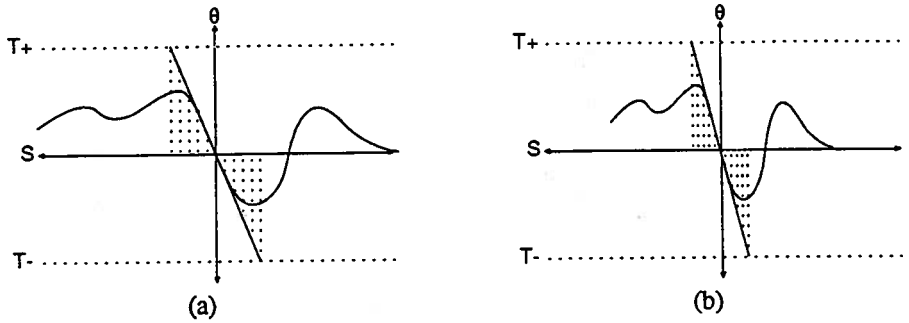


Figure 2: This figure illustrates the process of computing a shape vector. In (a) is shown an unscaled θ-s contour, and in (b) the same contour is scaled. In each case, the values of θ where the dotted vertical lines intersect the θ-s curve form the components of the shape vector. The sampling interval is determined by the points where the fitted line intersects the lines $\theta = \pm T$.

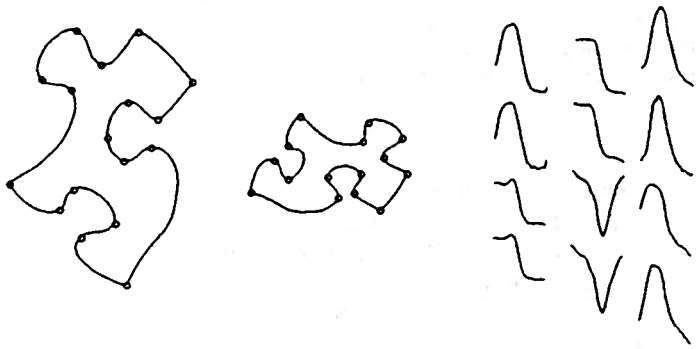


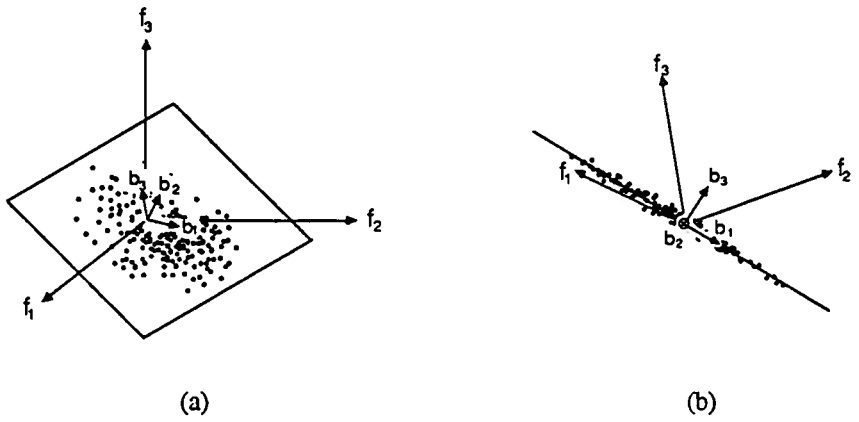
Figure 3: Each pair of shape vectors is comprised of elements from corresponding critical points from two images of objects. Of each pair, the shape vector on the top is from the image of the larger version of the objects, while that on the bottom is from the image of the smaller. The objects differed in scale by about a factor of 2. The scaling was accomplished by simply changing the distance of the camera to the objects.

have overcome this problem by setting a maximum allowable *average* residual squared error per sample and growing the region that is included in the fit symmetrically out from the critical point until the error per sample exceeds the limit. The algorithm then uses the last fit that did not exceed the limit. This yields a fit that uses only the approximately linear portion of the θ-s curve regardless of the scale. The strictness of the criterion of "approximate linearity" can be controlled by the size of the limit on the maximum allowable error per sample.

Once the slope of $\theta(s)$ at the critical point, i.e., at $s = 0$, has been estimated by above procedure, the halfwidth of the sampling interval $[-\delta, \delta]$, δ, is given by $\delta = T/m$, where m is the slope, and T is a parameter that controls how much of the θ-s curve is included in the sampling interval, independent of scale. Fig. 2 illustrates the preceding discussion of the computation of shape vectors.

The constant T above gives precise control over the degree of spatial locality exhibited by the shape vector. As described earlier, it also gives control over how informative the shape vectors, and therefore SICPN's, on average, will be for a particular set of objects. If T is made very small, then the shape vector will essentially be a sampling of a constant slope line, and, while being very spatially compact, it will contain essentially no information. T can be made progressively larger until the shape vector becomes a sampling the shape of a large part of the contour of an object, and is therefore very informative. Unfortunately, such a feature would be too global to be of use in a robust 3-d object recognition algorithm. Other factors make it inadvisable to make T too large, primarily the increased sensitivity to errors in the location of the critical point, thus reducing the noise resistance and effective informativeness of the feature.

The M-dimensional shape vector we have obtained above is itself a good feature possessing many of the properties that we enumerated above. However, it can be further improved. The vector has been degraded by a number of distortions and noise, most of which are high-frequency in nature. Thus, keeping the informative low-frequency part while discarding the noisy high-frequency part will improve the feature. In addition, the sample points contain much redundant information



(a) (b)

Figure 4: A hypothetical 3-d feature space is shown in (a) whose elements are highly correlated among themselves (dots represent features in the space). The correlation can be seen from the degree to which the features cluster near the plane. Shown in (b) is a different view of (a) looking with $\mathbf{b}_2$ going into the paper. The vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ span the plane while $\mathbf{b}_3$ is orthogonal to it. When represented in terms of the basis $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$, the $\mathbf{b}_3$ component of the vectors will be near zero and may be ignored, achieving a degree of data reduction. The K-L expansion allows such a basis to be computed.

due to the continuity of the θ-s curve from which they were derived. In order to improve the compactness and the noise properties of the shape vector, it is projected onto a set of orthogonal vectors, much like a Fourier decomposition. This set of orthogonal vectors is found by applying the Karhunen-Loève (K-L) expansion to a large set of representative feature vectors. A complete description of the theory behind the K-L expansion can be found in many texts on probability, statistics, and random processes, such as [16]. We will give a geometric explanation of what it accomplishes. Given a set of M-dimensional feature vectors, such as the local shape vectors described above, the K-L expansion yields a set of orthonormal M-d vectors which are ordered in the following manner: the first vector is chosen to lie in the direction of the largest variance in the set of feature vectors. The second will lie in the direction of the largest data variance in the $(M-1)$-dimensional subspace that is orthogonal to the first, and so on. Directions that contain little of the variance in the data can contain little information. Thus, only the components of the feature vector in the high-variance directions are important, while the others can be discarded without significantly degrading the informativeness of the feature. This leads to features that are more compact. The dimensionality of the subspace of $\Re^M$ onto which the features are projected is chosen to give a desired degree of information content versus compaction. If the subspace has dimensionality M, then no information is lost, but there is no compaction (and no reason to perform the K-L expansion). Typically, as will shown experimentally in the following section, the dimensionality of the subspace can be much smaller than M and still have most (95% or more) of the variance of in the data. Fig. 4 illustrates geometrically how the directions are chosen in a hypothetical 3-d feature space, and how the data compression results.

We have now described how a SICPN is computed. We still have not shown explicitly that SICPN features possess the properties that we have argued are necessary for a feature to be useful for 3-d recognition. That SICPN's possess some of these properties can be inferred from the algorithm for computing the SICPN's. These properties include:

1. Invariance to certain viewing parameters, namely rotation, translation, and scaling; insensitivity to scene lighting parameters;
2. Ease of segmentation, amounting to an edge detection followed by critical point detection;
3. Possession of a compact representation (as will be shown in the following section, applying the K-L expansion to the shape vectors typically compresses the amount of data needed to represent them by an order of magnitude).

Of the remaining properties that we have argued are desirable for a feature to possess, specifically:

(a) noise resistance;

(b) informativeness;

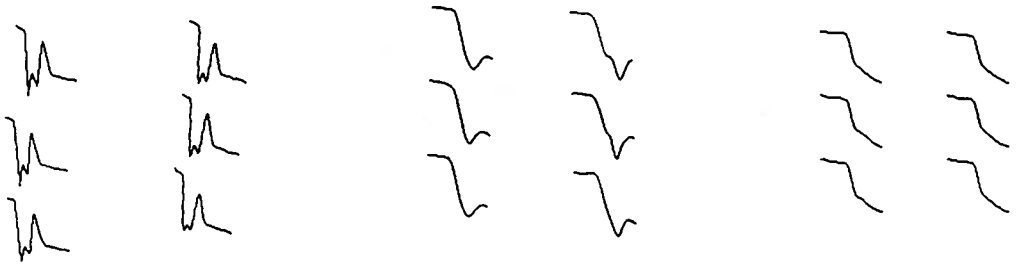


Figure 5: Shown are SICPN shape vectors. These shape vectors were computed from images of the plastic bottle whose edge contours are shown on the left hand side in Fig. 6. The shape vectors in each block are derived from the neighborhoods of corresponding critical points. The top left shape vector the column is computed from the smallest contour, the one below it from a contour that is a factor of $2^{\frac{1}{4}}$ larger, and so on to the largest in the right bottom of each block. The value of the parameter T was set at 2.5 radians per sample.

(c) minimization of the number of image-feature to model-feature correspondences that must be made during the recognition process;

the first two, (a) and (b), would be difficult or impossible to prove using realistic assumptions. We will therefore demonstrate them empirically. The last property, property (c), is only applicable to compound features that are constructed out of spatially local primitives. This criterion is not directly applicable to SICPN's. However, compound features using SICPN's as subparts will possess this property if the SICPN's themselves are informative.

5 Experimental Characterization of SICPN's

In this section, we will accomplish three tasks:

- Empirically demonstrate the invariance properties of SICPN's via plots of sets of shape vectors.
- Give an example of the result of applying the K-L expansion to a set of features from images of four objects at several scales.
- Most importantly, quantify the informativeness of SICPN's on a set of real objects.

The images that were used in these experiments were 480 × 480 × 8 bits. Canny's edge detector was applied using $\sigma = 3.0$ pixels. Samples on the θ-s curve were chosen to be approximately the width of one pixel (in image coordinates) of arclength apart. The critical points were detected with the multi-scale method described above. Three scales were used with ratios of 1.5 between them. The width of the smallest Gaussian was $\sigma = 4.0$ samples (of the θ-s curve).

Fig. 5 empirically demonstrates the invariance properties of the shape vectors. Shown there are 51-dimensional shape vectors computed from images of a plastic bottle (shown on the left of Fig. 6). The bottle appeared at random translations and orientations, while the scaling of the bottle differed by factors of $2^{\frac{1}{4}}$. The scaling of the images was achieved by adjusting a zoom lens on the camera. From the figure, the similarity of corresponding shape vectors, even at widely varying scales, is evident. Since real data is being used, clearly the invariance is not perfect; noise and other distortions degrade the ideal performance. Later in this section, we will attempt to more precisely measure the degradation.

Next, in Fig 7, we show ten orthonormal vectors resulting from application of the K-L expansion a large set of shape vectors derived from edge contours taken at 4 or 5 scales (each again differing by a factor of $2^{\frac{1}{4}}$ from each other) of four objects: the plastic bottle, a baby's rattle, and two jigsaw puzzle pieces, shown in Fig. 6. These ten vectors are those which contain the largest fraction of the variance in the set of shape vectors. In this example, the shape vectors were again computed using a value of $T = 2.5$ radians per sample. From the figure, it can be seen that only the first four of the orthonormal vectors need to be kept in order to retain more that 95% of the variance in the shape vectors. In the case of this example, the shape vectors consisted of 51 elements, but we need only to project them onto four directions, yielding a new vector of only four

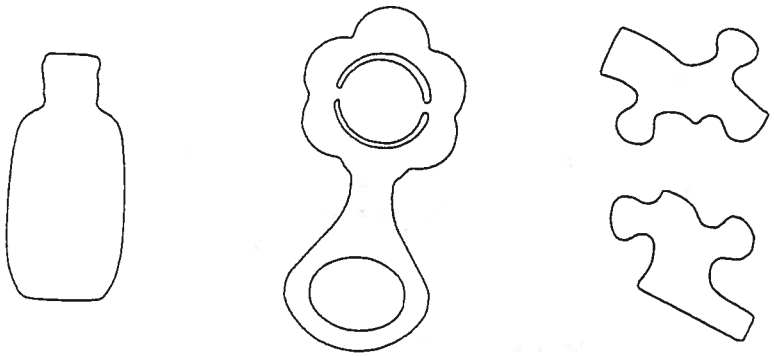


Figure 6: A plastic bottle, baby's rattle, and two jigsaw puzzle pieces used in experiments.

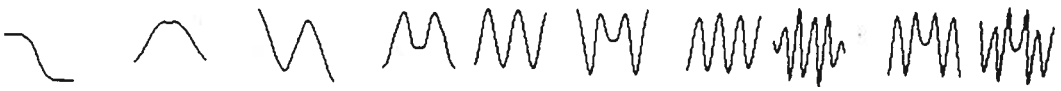


Figure 7: Shown are ten orthornormal vectors computed from a large set of shape vectors which included images four different objects at several different scales. The value of the parameter T was set to be 2.5 in this example. The shape vectors had 51 elements apiece. From left to right, the following figures are the percentage of the total variance in the set of shape vectors that was contained in the direction of that particular vector: 68.9%, 18.8%, 4.6%, 4.2%, 1.5%, .74%, .31%, .16%, .15%, .13%. Note that the first four vectors contain 96.7% of the variance. Therefore, the projection of a feature onto these four vectors will yield an encoding of the shape that has lost very little of the shape information contained in the original 51 element shape vector.

elements that has lost very little of the shape information in the original 51 element shape vector. This is a compression of more than an order of magnitude.

The set of vectors in Fig 7 also has the property that the high-variance directions tend to be lower frequency than the lesser variance directions. Since most of the noise and distortion in the shape vectors is high-frequency in nature, the projection operation performs a useful filtering operation in addition to data compression.

Finally, we turn to the problem of characterizing the informativeness and noise resistance of SICPN features.

We first propose a method of quantifying the informativeness of a set of feature vectors over a set of 2-d objects. We have restricted our attention to 2-d objects for the following reason. SICPN's are based only on the local shape of an object, and the local shape, in turn, may vary if an object is rotated about an axis that is not perpendicular to the image plane. This is an unavoidable phenomenon that has two causes. The first is projective distortion. It is possible, using the theory of differential and algebraic invariants [15], to arrive at invariants of arbitrary space curves with respect to *all* of the viewing parameters involved in projection, be it parallel projection or perspective projection. However, such invariants are not very useful due to the second phenomenon responsible for shape variations: object self-occlusion. This phenomenon makes it impossible to compute features that are invariant to such rotations for general 3-d objects.

Let $\mathcal{F}$ be the set of all SICPN features of dimension N. Define the equivalence class C_i to be the set of all features $f \in \mathcal{F}$ such that they have the i^{th} critical point at their centers. Ideally, the features $f \in C_i$ will be identical regardless of translation, rotation and scale. In practice, of course, there will be noise. A measure of how informative our features are is how little, on average, they vary within the classes C_i versus how distinctive they are between classes C_i and C_j where $i/neqj$. A measure of how the feature vectors vary within their class is simply the vector of standard deviations of the components. Call this vector d, with components d_i, $i = 1...N$. A measure of how distinctive the features are is the vector of standard deviations computed over all $f \in \mathcal{F}$. Call this vector D, with components D_i $i = 1...N$. Finally, taking the element by element ratios of d and D, we obtain the vector I with components $I_i = \frac{D_i}{d_i}$, $i = 1...N$. This vector measures the interclass distinctiveness versus intraclass variation for each component of a feature. If the classes C_i are viewed as subclusters in a

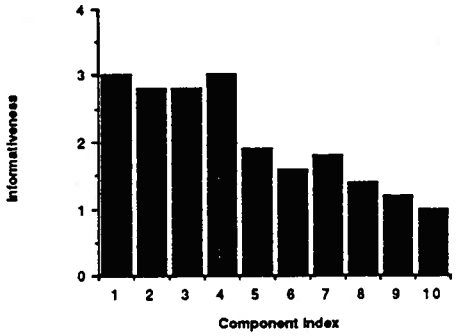


Figure 8: Plotted are the components of the vector I for :PN vectors. A value of $T = 2.5$ was used. Images of the objects shown in Fig. 6 were used. The scale of the objects varied over a range of $2^{2.25}$.

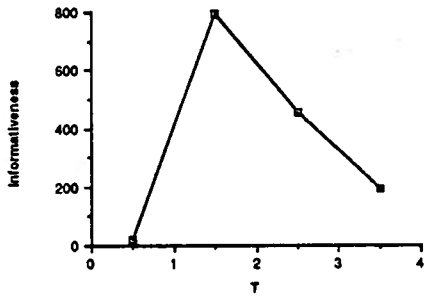


Figure 9: The effect of increasing the value of the parameter T on the informativeness $\mathcal{I}$ for five element SICPN's is plotted in the graph. These SICPN's were computed only from images of the plastic bottle.

larger blob of features situated in $\Re^N$, then it is of the components of I is the ratio of the volume of the entire blob of features to the volume of an average single-class subcluster. This number, $\mathcal{I} = \prod_{i=1}^{N} I_i$, is the measure of informativeness that we will use. The larger $\mathcal{I}$ is, the easier it will be to distinguish the classes from each other. This also corresponds to our intuitive notion of informativeness.

Fig. 8 plots the components of I for the case of 10-d SICPN's. The SICPN's were again computed from the four objects shown in Fig. 6. The shape vectors are computed again using the value $T = 2.5$. The shape vectors were then projected onto the orthonormal set depicted in Fig. 7. The images were taken over a range in scale of a factor of $2^{1.25}$. Two images of each object were taken at intervals of scale that had ratios of $2^{.25}$. The translation and orientation parameters of each object at each scale were random. As expected, the ratios tend to decrease as the index of the component increases. This is an effect of the K-L decomposition.

Fig. 9 plots $\mathcal{I}$ for five component SICPN's as a function of T. As can be seen, the informativeness increases as the T is increased. As mentioned earlier, this is not free since as T increases, the spatial locality of the feature is made proportionately less.

6 Summary and Future work

In the course of this work, we had one primary goal: to find good features for shape-based, recognition of objects from 3-d intensity images. We found that features used in other research for this task were not general enough to be used to recognize completely general objects. Therefore, we searched for better set of features that could augment or replace features used in previous 3-d recognition systems. In particular, we searched for spatially local features that are informative, easily segmented, and robust. The results of the search are the scale-invariant critical point neighborhood (SICPN) features we have described here. We have argued that SICPN's possess many of the properties that good features for 3-d shape-based recognition should have. In addition, we have attempted to quantify and measure the informativeness of SICPN's, arguably the most important property of a good feature.

For all of their advantages, SICPN's are not ideally suited to all types of objects, in particular, objects that have few corners and many lines. We feel that using both types of features, linear segments and SICPN's, will provide robustness in a 3-d recognition system.

In the future, we are interested in alternatives to the multi-scale critical point detector. While it is a vast improvement over a single-scale detector, we feel that other methods may work even better and provide a more sound theoretical foundation. In particular, a method based on the work of Fischler and Bolles *et al.* [5] is of interest.

References

[1] R. C. Bolles and R. A. Cain, "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method," in *Robot Vision*, A. Pugh, Ed., 1984.

[2] R. A. Brooks, "Symbolic Reasoning Among 3-d Models and 2-d Images," *Artificial Intelligence Journal*, Vol. 17, pp. 285-348, 1982.

[3] C. H. Chien and J. K. Aggarwal, "Shape Recognition from Single Silhouettes", *IEEE First International Conference on Computer Vision*, pp. 481-490, 1987.

[4] C. I. Connolly, J. L. Mundy, J. R. Stenstrom, D. W. Thompson, "Matching From 3-D Range Models Into 2-D Intensity Scenes," *Proceedings of the IEEE First International Conference on Computer Vision*, IEEE Cat. No. 87CH2465-3, pp. 65-72, 1987.

[5] M. A. Fischler and R. C. Bolles, "Perceptual Organization and Curve Partitioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 1, pp. 100-105, January 1988.

[6] C. Goad, "Special Purpose Automatic Programming for 3-d Model-Based Vision", *Image Understanding Workshop*, pp. 94-103, June 1983.

[7] P. G. Gottschalk, J. L. Turney, and T. N. Mudge, "Two-Dimensional Partially Visible Object Recognition Using Efficient Multidimensional Range Queries," *Proceedings of the 1987 IEEE Conference on Robotics and Automation*, IEEE Cat. No. 87CH2413-3, pp. 1582-1589.

[8] Radu Horaud, "Methods for Matching 3-D Objects with Single Perspective Views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 3, pp. 401-412, 1987.

[9] Daniel P. Huttenlocher and Shimon Ullman, "Object Recognition Using Alignment," *Proceedings of the IEEE First International Conference on Computer Vision*, IEEE Cat. No. 87CH2465-3, pp 102-111, 1987.

[10] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson, "Object Recognition by Affine Invariant Matching," *Proceedings of the 1988 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Cat. No. 88CH2685-4, pp. 335-344, June 1988.

[11] Y. Lamdan and H. J. Wolfson, "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," *Technical Report No. 368: New York University*, 1988.

[12] D. G. Lowe "Three-Dimensional Object Recognition from Single Two-Dimensional Images", *Artificial Intelligence Journal*, Vol. 31, pp. 355-395, 1987.

[13] Schunck, B. G. "Gaussian Filters and Edge Detection," *General Motors Research Publication No. GMR-5586*, pp. 33-47, October 20, 1986.

[14] D. W. Thompson and J. L. Mundy, "Three-dimensional Model Matching From an Unconstrained Viewpoint," *Proceedings of the 1987 IEEE Conference on Robotics and Automation*, IEEE Cat. No. 87CH2413-3, pp. 208-220.

[15] I. Weiss, "Projective Invariants of Shapes," *Proceedings of the 1988 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Cat. No. 88CH2685-4, pp. 291-297, June 1988

[16] E. Wong, *Introduction to Random Processes*, New York: Springer Verlag, 1983.