

# Generalized Shape Autocorrelation

Andrea Califano and Rakesh Mohan

IBM, Thomas J. Watson Center  
Exploratory Computer Vision Group  
P.O. Box 704, Yorktown Heights, NY 10598

## Abstract

This paper presents an efficient and homogeneous paradigm for automatic acquisition and recognition of nonparametric shapes. Acquisition time varies from linear to cubic in the number of object features. Recognition time is linear to cubic in the number of features in the image and grows slowly with the number of stored models. Nonparametric shape representation is achieved by spatial autocorrelation transforms. Both acquisition and recognition are two-step processes. In the first phase, *spatial autocorrelation* operators are applied to the image data to perform local shape analysis. Then, spatial autocorrelation operators are applied to the local shape descriptors to either create entries (acquisition) or index (recognition) into a table containing the distributed shape information. The output of the table is used to generate a density function on the space of possible shapes with peaks corresponding to high confidence in the presence of a particular shape instance. The behavior of the system on a set of complex shapes is shown with respect to occlusion, geometric transformation, and cluttered scenes.

## Introduction

This paper focuses on the issue of a scalable and homogeneous approach to the acquisition and extraction of generalized, nonparametric shapes from images. By scalable, we mean a technique which is tractable, and possibly computationally efficient, with respect to increasing sizes of images, increasing number of shape models stored and an increasing complexity of the models themselves.

General surveys on model-based object recognition can be found in [BJ85, CH86]. Most recognition paradigms, lack a corresponding framework for automatically acquiring object shapes [BCK89, BHH83, Br83, Go83, HH87, IK88]. The object models used, rather than being acquired from raw data, are either handcrafted, or generated assuming that precise geometric information is available.

A few approaches, e.g., geometric hashing [KS86, LW88] have been successfully used for both perceptual tasks, however they do not easily generalize to complex shapes and 3D objects. Often, this is a consequence of using recognition clues which are not invariant with respect to scale, and thus requires analysis at multiple level of resolution [Bu88]. As a consequence, a homogeneous shape storage-recall mechanism which is capable of dealing with increasing levels of complexity, both of the shape and of the scene, is still an open issue.

The approach presented here addresses these concerns and

offers the basis for a homogeneous, scale invariant, and tractable approach to the problem of acquiring and recognizing shapes from imagery. The generality of the formulation allows to extend this approach to a variety of domains.

The method exploits *long distance correlation* information in shapes to extend the concept of parameter transforms, used for the extraction of simple parametric features, such as lines or circles [Ho62, DH72], to a much more powerful and general paradigm for the extraction of arbitrary non-parametric features. This is achieved much in the spirit of the original generalized parameter transform [Ba81b], but without any of the limitations which make that approach impractical.

A two stage process is proposed, see Fig 1. First, short range autocorrelation operators are used to map from the image into a small set of simple local shape descriptors. A global autocorrelation operator is devised to map these local descriptors, distributed across the image, into index vectors. These are used as indices into a look-up table which contains the shape model representations.

During acquisition, entries are added to the table based on the index vectors generated by the transforms. For recognition, the same vectors are used to index into the table. The table output, in turn, generates a density function in the final shape space, where vectors correspond to specific instances of the stored shape models. Position, orientation and scale, as well as the symbolic label of the identified shape, are extracted simultaneously.

After each of the two symbolic parameter transforms, *constraint satisfaction* techniques [FB81, CBT89, Mo90] are used to select the best among incompatible local and global shape hypotheses. Only those with the best match to the input data survive, thus effectively segmenting the image with respect to the extracted feature set.

## Spatial Autocorrelation

We introduce spatial autocorrelation operators and describe how they can be successfully used in a generalized parameter transform framework. Some notions are recalled from previous work [Ca88, CBT89].

## The Generalized Neighborhood Framework

The Generalized Hough Transform, proposed by Ballard in [Ba81b], has shown how stochastic evidence integration techniques can be applied to the recognition of arbitrary

shapes in 2-space. However, due to the locality of the parameter estimation technique typical of the Hough paradigm, the complexity of the model search space explodes exponentially once arbitrary rotation and scale transformation are allowed in the input. For this and other reasons the Generalized Hough Transforms has failed to gain wide acceptance as a standard technique for shape extraction.

Recent work [Ca88, CBT89] has shown how it is possible to further generalize the notion of parameter transform to include a mechanism for fusing evidence embedded in distant portions of the image. This is achieved by extending the neighborhood concept to include compact nonconnected data set (Generalized Neighborhoods) and by devising transform operators for this new sparse data structure. Parameter transforms based on Generalized Neighborhoods typically achieve one to two orders of magnitude better accuracy in the parameter estimates of simple parametric features [Ca88]. They have been used to reliably extract up to 8-dimensional parametric features, such as conic sections in 3-space from range data [CBT89].

In the usual parameter transform formulation, a local mapping operator,  $f(\mathbf{P}, x, y)$ , is devised to estimate the likelihood of the presence of a specific feature  $\mathbf{P}=(p_1, p_2, \dots, p_s)^T$ , in a small local neighborhood  $P(x, y)$  in the image space, centered around the point  $(x, y)$ . Here, the parameters  $p_1, p_2, \dots, p_s$  uniquely identify the feature of interest. Evidence integration is performed by integrating the estimator over the entire image

$$E(\mathbf{P}) = \int_{Image} f(\mathbf{P}, x, y) dx dy$$

The resulting value is a confidence measure on the presence of the corresponding feature in the input. By repeating the operation for each possible feature, a density function over the parameter space of the features is generated. Peaks in the density function indicate high values of confidence with respect to the corresponding feature. Clustering technique are used to isolate the peaks from the background noise.

The mapping operator, in the case of Generalized Neighborhoods, is structured as a spatial autocorrelation function of the form:

$$f(\mathbf{P}, x, y) = \int \dots \int [f(\mathbf{P}, x, y, \dots, x_k, y_k)]^2 dx dy \dots dx_k dy_k$$

$A(\{(x_i, y_i)\}) A(\{(x_i, y_i)\})$

where  $k$  determines the order of the correlation function.  $A(\{(x_i, y_i)\})$  is a neighborhood for partial evidence integration which can depend on the value of the different  $(x_i, y_i)$ . Such

neighborhood, usually, can be the entire image or a portion of the image centered around  $(x, y)$ .

Specific advantages of autocorrelation mapping operators are discussed in [CBT89]. Here we would like to mention increased accuracy of the transform (up to two orders of magnitude), the ability to use partial forms of the mapping operator to link features to image regions using a probabilistic measure, and the increased hyper-volume under the density function in parameter space. This final feature allows for the successful implementation of high dimensional parameter transforms.

### Non Parametric Feature Extraction

Although the term ‘‘parameter transform’’ might seem to indicate otherwise, it is possible to use this technique for nonparametric feature extraction. By nonparametric features, we simply mean features that can not conveniently be described by a small set of parameters. It is known from basic analysis that any piece-wise invertible 2D-curve can be approximated with arbitrary precision by its Taylor series. So, given the usual finite precision requirement, many curves can be locally approximated with sufficient accuracy in terms of the first few terms of the series. Even closed or disconnected curves can have relatively simple polynomial representation [Ta88]. However, retrieving the curve parameters from an image can be an extremely hard task which almost invariably requires optimal segmentation of the input.

Besides, the existence of a parametric representation does not guarantee its stability. In other words, small variations in some of the parameters, given the nonlinearity of Taylor series, can produce arbitrary large variations in the form of the associated 2D-curve and vice-versa. Thus matching the extracted representations to the stored ones becomes almost impossible. In these cases, parametric representations are not viable candidates for either the acquisition or the extraction process.

### Autocorrelation Shape Transforms

We provide a different approach to the issue of shape representation. Initially we restrict the analysis to the context of 2D-curves. Later, we generalize the framework to more complex domains.

Let us suppose that given a few nonparametric 2D curves we can uniquely identify a set of points on each curve and estimate their 0<sup>th</sup> and 1<sup>st</sup> order properties. We introduce a discrete spatial autocorrelation operator which will map

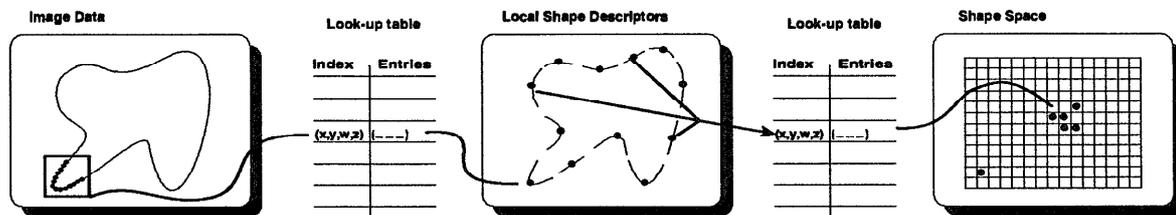


Fig. 1: System Architecture

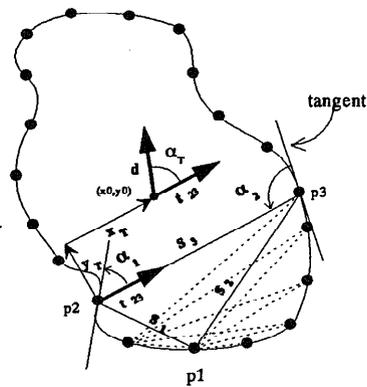


Fig. 2: Correlation of geometric information

these sets into a density function on a feature parameter space. Vectors in such a space uniquely identify position, rotation, and scale of the feature, as well as the feature itself with respect to others.

Each possible point in the image, drawn from the previously described sets, can be associated with all possible combinations of two other points, thus forming a set of triangles, see Fig. 2. For each such structure we generate a four dimensional index. This is built using the two ratios  $s_2/S$  and  $s_3/S$ , where  $S = s_1 + s_2 + s_3$ , which determine the geometry of the triangle, and the two angles  $\alpha_1$  and  $\alpha_2$  which describe 1<sup>st</sup> order properties of the curve with respect to the given 3-point set.

This index is scale and rotation invariant, if the position and properties of the points on the curves can also be determined independently of such transforms. In fact, if other local properties of the points are also invariant, they can be used to form longer indices, as we show later. Different quantization criteria are applied to each of the index parameters to form a set of discrete variables with a finite number of values.

During the acquisition phase, we consider a single, unoccluded 2D shape in the image so that the position of its center of mass  $(x_0, y_0)$ , its scale (e.g., perimeter)  $\mu$ , its orientation vector  $\mathbf{d}$ , and a symbolic label  $L$  are known. In the case of 3D shapes, different views corresponding to different aspects of the object [KD79,CF82] are acquired. Alternatively, given the high computational cost of finding different aspects of an object [GM90], and the ability of our system to recognize shapes from partial instances and with some amount of projective invariance, the Gaussian viewing sphere can be sampled at small regular intervals.

For each given triangle we compute a tuple containing four elements: the label of the object, the position  $(x_T, y_T)$  of the center of mass in the new right hand coordinate system defined by the normalized vector  $\mathbf{t}_{23}$  and its corresponding orthonormal one, the ratio  $\rho = \mu/S$ , and the angle  $\alpha_T$  between the two vectors  $\mathbf{t}_{23}$  and  $\mathbf{d}$ . All these parameters are also scale and rotation invariant, and are used to recover position, orientation and scale of a feature from the corresponding 3 point sets.

For each possible combination of three points over the

input data, we use the generated discrete index to select an entry in a look-up table, and we add the tuple  $\langle L, (x_T, y_T), \rho, \alpha_T, Count \rangle$  to it. Here *Count* is originally set to one. Every time a tuple produced for a given index exactly matches one already present in the table on the five initial parameters, *Count* is incremented by one. The significance of an entry is proportional to its *Count*.

During recognition, in a similar fashion, all possible combinations of 3 points from the image are used to generate similar sets of indices. For each tuple contained in the table at any given index, a feature-space parameter vector is generated which uniquely determines a possible shape. Based on the value of *Count*, the evidence density on the parameter space is updated. Triplets located on the same feature or shape correctly estimate the parameter vector and thus accumulate evidence in the density function. Other triplets produce pseudo-random results which will be scattered over the entire parameter space and result in negligible accumulation.

For each parameter vector in the feature-space, we record the points used to generate its value, and the number of times they voted for it. This information is then used to probabilistically associate regions in the image with feature vectors in parameter space as discussed in [CBT89]. Constraint satisfaction techniques [Ba81a,CBT89,Mo90] are used to filter the result of the transforms based on this knowledge. Different vectors generated by overlapping data-point subsets compete in a "winner-take-all topology" [FB81], so that only the one with highest evidence support from the image is selected. Since the selected features do not share data support, they segment the original input image.

By modifying the number of points, properties, and relationships considered, the same framework of evidence integration for nonparametric shapes can be extended to many different domains. In three-space, for instance, one could use the relative position of any three points from a range data image, with their 1<sup>st</sup> order properties (surface normals) represented with respect to the plane defined by the three points.

## Local Shape Descriptors

Stable and robust behavior of the transform depends on the correct selection of the set of points on the 2D-shape. No known data-driven techniques guarantee consistence in the selection of points, for a given shape, across different scales and in presence of quantization noise typical of digitized images. One solution is to exhaustively use all possible edge points. This corresponds to the previous formulation in terms of spatial autocorrelation transforms, see previous 2 equations. The mapping operator, in this case, has the discrete structure of table look-up, and the integral is replaced by a summation over subsets of the image discrete locations. If the image used for shape acquisition contains enough data points, this would ensure the overlap of an extensive data subset even in presence of noise and input transformation on scale, rotation and translation.

However, the  $O(n^3)$  computational complexity with respect to the number of considered points makes the ap-

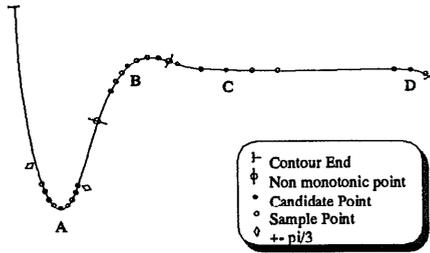


Fig. 3: Sampling on a curve

proach unappealing when large data sets are explored. To overcome this difficulty, the shape transform is split into two distinct phases. In the first one, we encode the local 2D shape information into a small set of localized and symmetric shape descriptors. Next, we use this intermediate representation as the data set for the transform that recovers global shape information.

We now have some other invariant parameters, namely those describing the local structure of a curve around each point in a given triplet. These are used to produce a higher dimensional vector for indexing in the look-up table, endowing the transform with even more selectivity. The framework for the acquisition and extraction of local shape descriptors is analogous to the one described in the previous subsection. While global shapes are not associated with any specific point on the shape and are positioned through their centers of mass, local shapes are associated with a specific point on the shape. For instance, elliptical arcs are associated with points where curvature reaches local extrema. Local shape descriptors considered are lines, circular arcs, and elliptic arcs (minima and maxima of curvature). Larger sets of local shape descriptors can also be considered [AB86].

First edges are extracted from images using standard techniques such as a Canny edge-detector [Ca86]. Edges are then linked into contours using simple eight neighbors connectivity. Given a point  $(x_0, y_0)$  on a contour, we symmetrically sample around that position along the curve (see Fig. 3), generating a set  $\{(x_i, y_i), -N \leq i \leq N\}$ . The sampling rate is proportional to the length of the longest symmetrical interval around the point for which the tangent behavior is monotonic on a coarse scale, and the total tangent variation is less than  $2\pi/3$ . It is also inversely proportional to the total variation in tangent angle on the symmetric interval if less than  $2\pi/3$ . We assume that faster variation in tangent correspond to smaller features which accordingly require finer sampling for detection. In this way, a quasi scale invariant sampling of the contours is achieved. Next we form all possible combinations of the point  $(x_0, y_0)$  with two others from the set  $\{(x_i, y_i)\}$ . The vector  $(s_1/S, s_2/S, \alpha_1, \alpha_2)$ , described in section 2.3, is computed from each triplet and it is used to index into the local-shape look-up table.

During acquisition, the computation is performed with  $(x_0, y_0)$  located at the center of symmetry for the considered shape. For each triplet, the tuple  $\langle \lambda, \beta, S, Count \rangle$  is inserted in the table. Here,  $\lambda$  is the symbolic label for the local shape.  $\beta$  is the angle between the vector  $t$  and the normal to the contour  $n$  required to recover the shape orientation from the

triplet.  $S = s_1 + s_2 + s_3$  is used for scale normalization. The role of parameter *Count* is identical to that explained in the previous subsection.

During recognition, given  $(x_0, y_0)$  and the sample set  $\{(x_i, y_i)\}$ , triplets are formed and indices computed in an analogous fashion. For each entry in the table, a vector in a local shape parameter space is computed which uniquely identifies the shape, rotation and scale. The density function on the parameter space is updated with the value of *Count*. After all the triplets have been considered, the feature vector with the highest data support describes the shape around  $(x_0, y_0)$ . By considering the subset of  $\{(x_i, y_i)\}$  that successfully voted for a given vector, it is also possible to recover the section of the contour associated with the corresponding shape descriptor. The process is repeated for the next point on the contour at the determined sampling rate, until the entire contour has been considered, and then for all other contours. Since sampling rates are recomputed for each point, sample density varies dynamically along the curve.

After the extraction phase, local descriptors supported by overlapping portions of the contours compete, leaving only the ones with the highest confidence. In this way, contours are naturally segmented into a small set of nonoverlapping localized shapes which allow us to recover most of the original information, see Fig 4, 5 and 6. The local descriptors are positioned at coarsely sampled contour locations. To increase stability, a finer localization is obtained by extracting new descriptors on a pixel by pixel basis, in a small neighborhood of the original local descriptors, and selecting again from among the results, the one with highest support. Through extensive experimental testing, this approach has shown a high degree of stability with respect to scale, rotation, translation, and limited projective transformations of the input data.

The final set of descriptors is used for the global shape extraction. The index, in this case, becomes  $(s_1/S, s_2/S, \alpha_1, \alpha_2, \lambda_1, \lambda_2, \lambda_3)$ , where  $\lambda_i$  is the symbolic label for the shape at point  $(x_i, y_i)$ . Scale information about the shapes could also be used since the normalization factor  $S$  is known for each triplet, but this has not currently been implemented.

## Computational Complexity

An important result is that recognition time is not exponential in the number of image features and grows slowly with the number of models stored in the database. This property is required in machine vision to deal with the large number of shapes necessary for the analysis of real scenes.

A direct implication of the use of third order autocorrelation functions, during acquisition and recognition, is  $O(n_e^3)$  time complexity, where  $n_e$  is the number of edge elements in the image. The use of local shape descriptors changes the complexity function into  $O(n_s^3)$  where  $n_s$  is the number of local shapes,  $n_s \ll n_e$ .

The following sections introduce techniques to reduce the time requirement to  $O(n_s)$ .

## Acquisition

Local shape descriptors are assigned a weight,  $w_r$ , proportional to their visual relevance. This measure uses the length of their support normalized with respect to the size of the model (this favors large features) and/or the tangent angle variation per unit length that they describe (this favors corners). Larger features have a higher chance of being correctly identified from the input. Rapid variations in tangent correspond to curvature maxima which are well localized on contours.

For each model, it is possible to isolate a small constant number,  $c$ , of such highly relevant descriptors  $D_r = \{D_i : i \leq c\}$  from the others  $D = \{D_j : j \leq n_s\}$ ,  $D_r \subseteq D$ . We then generate all possible triplets with the first element from the set  $D$  and the other two from the set  $D_r$ . The total number of triplets formed,  $n_s \binom{c}{2}$ , is a linear function of  $n_s$ . Thus, the time required to acquire a shape model, or one view for a 3D object, is  $O(n_s)$ . This representation is still highly redundant as each feature is present in at least  $\binom{c}{2}$  triplets.

## Recognition

Since real world objects have in general a compact structure, it is possible to exploit this property to reduce the search space. For each identified descriptor we generate a circle of coherence centered at its location. The radius of this circle is proportional to the size of the local shape. Other descriptors from the same shape are more likely to be found within such circle. This technique has been demonstrated to reduce the cubic explosion in the number of triplets to a linear one [CBT89] with respect to the image size.

For a given local descriptor, not all the visually relevant features used with it to acquire the model may be present in its radius of coherence. However, due to the highly redundant shape representation, this is usually sufficient to identify the correct match. To make recognition robust with respect to noise in cluttered environments, we extend the radius of coherence until a required constant number of descriptors with desired visual significance  $w_r$ , normalized with respect to the one of the considered descriptor, are found and we create triplets only with these. This search can be made efficient using multiresolution analysis techniques, or maintaining the features in a heap structure. In either case, maximum search time is within  $O(n \log n)$ .

For each triplet, all the entries of the indexed slot in the shape table have to be processed. The average number of entries processed is  $cQ$ ,  $c = 10^{-3}$ , assuming a table size of  $10^6$ ,  $Q$  approximately  $10^3$ , and an average of  $10^3$  triplets generated. Thus recognition time is  $O(n \log n Q)$ .

## Shape Database

The shape database is akin to associative memories [HA81] which usually employ distributed representations, in that each of the shape descriptions is not localized to a particular location but distributed over the memory. Such

holographic representation engenders fault tolerance to loss of parts of the memory. Shapes are represented as a collection of entries in the table and parameter vectors are used to index into it. Tables are implemented either as arrays, or as hash tables, depending on their size and sparseness.

New shapes are learned by performing spatial autocorrelation on presented instances from images. Generality and robustness, to allow for the correct recognition behavior in the presence of noise and small projective transforms of the input, are achieved by storing different instances of the shape, by quantizing the parameter vectors (indices), and by stochastic index perturbation mechanism during acquisition. That is, a small randomness (order of the index parameter quantization) is added to the index for the look-up table along each of its dimensions. For each image and object, the acquisition is repeated until the indices no longer hit empty cells in the look-up table with a significant rate.

Our model acquisition paradigm is basically a visual shape learning mechanism. As experimentally shown in the next section, this technique is stable to changes in orientation, translations, scale and even small variations in shape, such as those caused by different limited projective transforms. This capability to handle small shape variations not only makes it robust to noise but also allows for generalization of the learned models. Thus when new, unlearned shapes are analyzed, they are matched by the closest stored representation. If the presented shape is significantly different with respect to any of the stored ones, it is not classified. However, once the new shape has been acquired, different instances of this shape are correctly recognized. Thus, this learning system not only exhibits generalization capabilities but also separability.

## Experiments

We report some experiments illustrating the different capabilities of our model acquisition and recognition system. We selected the domain of leaf shapes to demonstrate the acquisition and recognition of complex nonparametric shapes. Two of the leaves from this set are shown in Fig. 4 and 7. In this and following figures we show the contours obtained by linking the edges. The contours have been smoothed with a Gaussian filter. The local shape descriptors detected on the first leaf are shown in Fig. 5 and 6.

**Robustness to noise and small variations in shape:** 7% Gaussian noise was added to 20% of the points on the contours. The local shapes detected on this noisy image are shown in Fig. 8. Recognition and location were successful although the number of supporting features dropped to about 50% of those found in the original image.

**Scaling:** An image of the leaf is taken at a shorter distance leading to projected shape of the leaf being 1.8 times larger than the model, Fig. 9. The recognition phase returns the correct label for the leaf and the correct scale factor. Fig. 10 shows the recognized model, scaled by the computed scale factor, and projected onto the image.

**Rotation and translation in the image plane:** An image of leaf # 1 which has undergone rotation and translation in the image plane is considered. The system correctly iden-

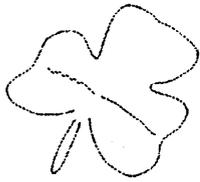


Fig. 4: Shape of leaf #1

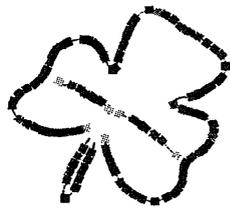


Fig. 5: Local shape descriptors

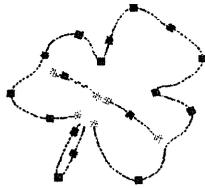


Fig. 6: Selected descriptors

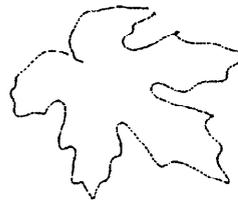


Fig. 7: Shape of leaf #2



Fig. 8: 7% Gaussian noise. Local shape descriptors

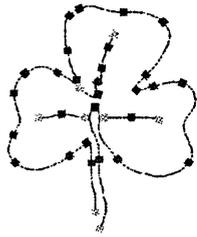


Fig. 9: Scaled instance. Local shape descriptors



Fig. 10: Recognition and location for Fig. 8

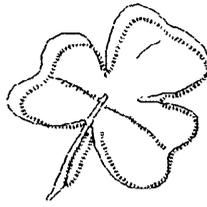


Fig. 11: Rotated and translated, recognition

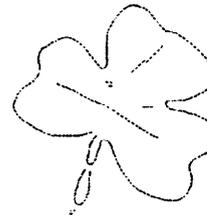


Fig. 12: Projective transform

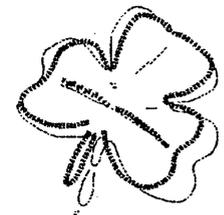


Fig. 13: Recognition and location for Fig. 12

ifies the shape and the parameters for rotation and translation and scale. The original leaf #1 model, translated and rotated with the recovered amounts is overlaid (Fig. 11). The difference in registration is due to the coarsely quantized computation of the geometrical transform parameters, e.g., the position of the center of mass. If needed, a more precise registration of the model to the image can be obtained by reducing the size of the quantization, rematching only the selected local shapes, and accumulating evidence only for the matched model.

**Viewpoint:** Slant and/or tilt of the object corresponds to viewing the object from different viewpoints. A leaf is viewed with the image plane slanted *and* tilted about 7% from a plane parallel to the leaf (i.e. projective transform, Fig. 12). There is also some magnification due to differences in the viewing distances. Even given the relatively large change in viewing direction, the correct model is recognized. This model with the scale and translation accounted for, is overlaid on the image in Fig. 13. Currently, recovery of viewing directions is not incorporated as the objects are 2D shapes. To show that these parameters can be recovered by learning the projected representations of the model at different viewing angles, we acquired the skewed leaf shape as a separate leaf model. In subsequent recognition of the skewed shape, the correct instance is recognized. This not only indicates the system can be incremented to handle 3D shapes, but the factors of *generalization* and separability, i.e. shapes are categorized as the most similar object in the shape memory but after learning of the shape, the system is able to distinguish between that similar shape and the newly learned one.

**Complex scenes:** We analyze an image, shown in Fig. 14, containing multiple, with some occluded instances of objects. The scene is cluttered with objects that have not been acquired. All the instances of leaf #1 are correctly recognized. Due to the presence of multiple objects, other models

in the memory also receive some votes. These models are suppressed by the constraint satisfaction mechanism. Again, the differences in the registration of the object model with the image (Fig. 14) reflect the quantized values used for computing the transforms.

## Conclusions and Future Directions

We have successfully demonstrated a system for the automatic acquisition and recognition of complex visual shapes which are not easily modeled by conventional parametric techniques. We have additionally demonstrated the use of spatial autocorrelation as a viable mechanism for describing such shapes. The associative memory structure used for learning and recalling these shapes exhibits the properties of robustness, generalization and recall from partial descriptions, that have usually been considered to be limited to neural mechanisms. Additionally, this holographic memory is efficiently implemented on conventional computers, both in terms of time and space requirements. The learning is accomplished in time complexity of  $O(n)$  to  $O(n^3)$ , where the model is described by  $n$  local shape descriptors. The size requirement for the memory is  $O(mn)$  to  $O(mn^3)$  where  $m$  is the number of distinct object shapes (or distinct object shapes times their different aspects for 3D shapes) and  $n$  the average number of local shapes per model. Recognition time is  $O(n_l Q)$  to  $O(n_l \log n_l Q)$ , where  $n_l$  is the number of local shape features in the image, and  $Q$  the number of models in the database.

Future work will explore interesting properties exhibited by this mechanism for supervised/unsupervised learning of perceptual concepts such as symmetry and hierarchical organization of models into subparts. A different line of research being pursued, includes stochastic techniques for

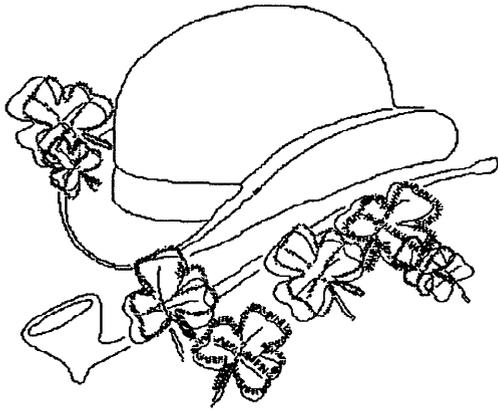


Fig. 14: Object recognition on complex scenes

applying autocorrelation operators in real-time tasks, such as in robotics, where concepts must be learned and recalled within short time frames.

### Acknowledgements

We would like to thank Ruud Bolle and Rick Kjeldsen for their reviews and comments and Norm Haas for help with the images.

### References

- [AB86] H. Asada and M. Brady 1986. The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1).
- [Ba81a] D.H. Ballard 1981a. Parameter nets: A theory of low level vision. In Proc. 7th Int. Joint Conf. On Artificial Intell., 1068-1078.
- [Ba81b] D.H. Ballard 1981b. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, Vol. 13, No. 2: 111-122.
- [BCK89] R.M. Bolle, A. Califano, R. Kjeldsen and R.W. Taylor 1989. Visual recognition using concurrent and layered parameter networks. In Proc. IEEE Conf. on Comp. Vision and Patt. Recognition.
- [BJ85] P.J. Besl and R.C. Jain 1985. Three-dimensional object recognition. *Computing Surveys*, Vol. 17, No. 1: 75-145.
- [BHH83] R.C. Bolles, P. Horaud, and M.J. Hannah 1983. 3DPO: A three-dimensional part orientation system. In Proc. 8th Int. Joint Conf. on Artificial Intell., 1116-1120.
- [Br83] R.A. Brooks 1983. Model-based three-dimensional interpretations of two-dimensional images. *IEEE Trans. on Pattern Analysis and Machine Intell.*, Vol. 5, No. 2: 140-150.
- [Bu88] P.J. Burt 1988, "Smart sensing within a pyramid vision machine," *Proc. of the IEEE*, Vol. 76, No. 8: 1006-1015.

- [Ca86] J.F. Canny 1986, "A computational approach to edge detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6: 679-698,.
- [Ca88] A. Califano 1988, "Feature recognition using correlated information contained in multiple neighborhoods," in Proc. 7th Nat. Conf. on Artificial Intell., pp. 831-836.
- [CBT89] A. Califano, R.M. Bolle, and R.W. Taylor 1989. Generalized neighborhoods: A new approach to complex feature extraction. In Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition.
- [CF82] I. Chakravarty and H. Freeman 1982. Characteristic view as a basis for three-dimensional object recognition. In Proc. SPIE Conf. on Robot Vision, 37-45.
- [Ch86] R.T. Chen and C.R. Dyer 1986. Model-based recognition in robot vision. *ACM Computing Surveys*, Vol. 18, No. 1: 66-108.
- [DH72] R.O. Duda and P.E. Hart 1972. Use of the Hough transform to detect lines and curves in images. *Comm. ACM*, 15(1): 11-15.
- [FB81] J.A. Feldman and D.H. Ballard 1981. Connectionist models and their properties. *Cognitive Science*, Vol. 6: 205-254.
- [GM90] Z. Gigus and J. Malik 1990. Computing the aspect graph for line drawings of polyhedral objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 2.
- [Go83] C. Goad 1983. Special purpose automatic programming for 3D model-based vision. In Proc. DARPA Image Understanding Workshop, 94-104.
- [HH87] C. Hansen and T. Henderson 1987. CAGD-Based computer vision. In Proc. Workshop on Comp. Vision, 100-105.
- [HA81] G. Hinton and J. Anderson 1981. *Parallel models of associative memory*. Lawrence Erlbaum Ass.
- [Ho62] P.V.C. Hough 1962. Methods and Means for Recognizing Complex Patterns, U.S. Patent 3069654.
- [IK88] K. Ikeuchi and T. Kanade 1988. Automatic generation of object recognition programs. *IEEE Proceedings*, Vol. 76, No. 8: 1016-1035.
- [KS86] A. Kalvin, E. Schonberg, J.T. Schwartz, M. Sharir 1986. Two-dimensional, model-based, boundary matching using footprints. *The International Journal of Robotics Research*, Vol. 6, No. 4.
- [KD79] J. Koenderink and A. van Doorn 1979. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, Vol. 32: 211-216.
- [LW88] Y. Lamdan and H.J. Wolfson 1988. Geometric hashing: a general and efficient model-based recognition scheme. In Proc. 2nd Intl. Conf. on Comp. Vision.
- [Mo90] R. Mohan 1990. Constraints satisfaction networks for computer vision. *Progress in Neural Networks*, O. Omidvar ed., Ablex.
- [Ta88] G. Taubin 1988. Nonplanar curve and surface estimation in 3-space. In Proc. of IEEE Conf. on Robotics and Automation.