# The feature transform for ATR image decomposition

Davi Geiger, Robert Hummel, Barney Baldwin, Tyng-Luh Liu, and Laxmi Parida

Courant Institute, NYU 251 Mercer Street New York, NY 10012

**Abstract**

We have developed an approach to image decomposition for ATR applications called the "Feature Transform." There are two aspects to the Feature Transform: (1) A collection of rich, sophisticated feature extraction routines, and (2) the orchestration of a hierarchical decomposition of the scene into an image description based on the features. We have expanded the approach into two directions, one considering local features and the other considering global features.

When studying local features, we have developed for (1) corner, T-junctions, edge, line, endstopping, and blob detectors as local features. A unified approach is used for all these detectors. For (2), we make use of the theory of Matching Pursuits and extend it to robust measures, using results involving $L^p$ norms, in order to build an iterative procedure in which local features are removed from the image successively, in a hierarchical manner.

We have also considered for (1) global shape features or modal features, i.e., features representing the various modes of the models to be detected. For (2) A multiscale strategy is used for moving from principal modes to secondary ones.

The common aspect of both directions, local and global feature detection, is that the resulting transformations of the scene decomposes the image into a collection of features, in much the same way that a discrete Fourier Transform decomposes an image into a sum of sinusoidal bar patterns. With the Feature Transform, however, the decomposition uses redundant basis functions that are related to spatially-localized features or modal features that support the recognition process.

## 1 Introduction

By viewing the decomposition of a sensor image as a feature extraction process, the ATR recognition problem can be seen as a pattern matching task of finding correspondences between model features and observed scene features. If the features are made invariant to geometrical transformations, then the recognition algorithm requires only that we find a match between a pattern of features for a model and a subset of the features of the image. Geometric hashing forms a method for performing the pattern matching task. Generally, pattern matching uses a collection of points in a single Euclidean space. However, the decomposition of an image will discover different kinds of features, such as corners, lines, edges, endpoints, and blobs. Some features are described by two or three parameters, and other features require a dozen parameters. Accordingly, patterns will be elements in a Grassmannian algebra, meaning a collection of points in different dimensional spaces. The geometric hashing [14] engine can also be used for such generalized patterns, but requires that the patterns be extracted stably from images, even in the presence of partial occlusion, overlapping features (such as corners and edges sharing common pixels), noise, and viewpoint transformations. Stability of the feature extraction process means, for example, that if a corner detector responds to a particular location on a vehicle, then it should provide the same information under a range of circumstances, such as viewpoint variations and varying operating conditions. For automatic target recognition, the features need to be rich, descriptive, discriminative, and stable.

In this paper we develop an approach to image decomposition for ATR applications called the "Feature Transform." There are two aspects to the Feature Transform: (1) A collection of rich, sophisticated feature extraction routines, and (2) the orchestration of a hierarchical decomposition of the scene into an image description based on the features. We have expanded the approach into two directions, one considering local features and the other considering global features.

When studying the approach based on local features we have developed for (1) corner, T-junctions, edge, line,



0-8194-1837-4/95/$6.00

endstopping, and blob detectors as local features. A unified approach is used for all these detectors. For (2), we make use of the theory of Matching Pursuits in order to build an iterative procedure in which local features are removed from the image successively, in a hierarchical manner. In particular a robust pursuit strategy is devised using results involving $L^p$ norms.

When studying the approach based on global features we considered for (1) shape or modal features, i.e., features representing the modes of the models to be detected. We derive the modes of the objects by concentrating on the profile of objects from different views. For (2), a multiscale strategy is used for moving from principal modes to secondary ones.

The resulting transformation of the scene decomposes the image into a collection of features, in much the same way that a discrete Fourier Transform decomposes an image into a sum of sinusoidal bar patterns. With the Feature Transform, however, the decomposition uses redundant basis functions that are related to spatially-localized features or modal features that support the recognition process.

The organziation of the paper is as follows. We first present a novel model for detection of local features such as corners, T-junctions, blobs. We then present a robust Matching Pursuit strategy, first extending the use to the features described above and then showing how it can be robustify by using results involving $L^p$ norms, that decomposes an image into its features by iteratively removing the "best" feature. We apply this method to images, but we have not yet integrated this module with the local features described above. We then show how to build a global feature representation of the images via the modes of the objects by concentrating on the profile of objects from different views. We finally show a strategy for decomposing an image into global feature modes, by focusing within a multiscale space from the principal mode to secondary ones. Some results using ATR images are demonstated. The unification of the global feature strategy with the local one is a subject of current investigation.

# 2 Local piecewise constant feature detection

A critical component of most recognition systems is stable, representative, feature extraction from sensor data. One of the key features to recognition are junctions, like corners and $T$-junctions, $Y$-junctions and so on. Such points, for example, coincide with the images of trihedral vertices of an object. These are critical features for recognition as suggested in early works by Huffman [11], Guzman [10] on line labeling. We develop a "junction detector," to finds corners, tri-corners (tri-junctions), quad-corners (quad-junctions), etc., defined as points where three or more homogeneous surface patches are located within an arbitrarily small neighborhood of the point. Thus, we are proposing an approach of fitting piecewise constant junction templates to a set of data. Unlike many previous approaches to junction extraction, our detector does not use prior edge extraction, but instead operates directly on the local image data. It differes from [6] in that we do find an optimal method to partition the image and can be extended to other features.

We model a junction as a region of an image where the values are piecewise constant in wedge-shaped regions emanating radially from a central point. Moreover, we require the partitions to be located such as to provide high contrast in the template values. The parameters of a junction consist of (i) the center location, (ii) the number of radial line boundaries, (iii) the angular direction of each such boundary, and (iv) the intensity value within each wedge. Using an $L^2$ norm and a contrast term we formulate the junction detection problem as one of finding the parameter values that yield a junction that best approximates the local data. The best-fit parameter values provide attributes of the detected junction. We use dynamic programming to excatly calculate the best-fit paramters. We also show how this approach can be applied to build blobs and end-stoping detectors.

## 2.1 Formulation of the Problem

We need to fit a template to the image at a point. The number of partitions, with intensity at each partition, defines a template. If the number of partitions is one, then the point in the image is a homogenous region with no junctions;

if two, the region has a corner[1]; if three, the region has a tri-corner[2] and so on.

Specifically, a template is defined in terms of the following parameters, $N$-the number of partitions, $\theta_p$-the angles where a partition occurs, where $p = 1, 2, ..., N$, and $T_p$-the template value for each partition. The data is given by $I(x, y)$. The formula for the $L^2$ error yields

$$E = \int\int [I(x,y) - T_{\{N,(\theta_p,I_p)\}}(x,y)]^2 g(x^2+y^2)\, dx\, dy,$$

where $g(x^2+y^2)$ is a monotonic decreasing function, e.g. $g(x^2+y^2) = \frac{1}{2\pi} e^{-(x^2+y^2)}$ and gives the scale of the junction detection. To further exploit the radial symmetry of the template (it does not depend on the radial coordinate) we choose the polar coordinate system

$$\rho = \sqrt{x^2+y^2}\,, \quad \theta = ArcTan\frac{y}{x}\,, \quad \text{and} \quad x = \rho\cos\theta\,, \quad y = \rho\sin\theta\,.$$

Thus, we have $\rho\, d\rho\, d\theta = dx\, dy$. In the new coordinate system the error can be written as

$$\begin{aligned} E &= \sum_{p=1}^{N} \int_{\theta_p}^{\theta_{p+1}} \int_0^\infty [I(\rho,\theta)^2 + T(\theta)^2 - 2I(\rho,\theta)\, T(\theta)]\, \rho\, g(\rho)\, d\rho\, d\theta \\ &= \sum_{p=1}^{N} \int_{\theta_p}^{\theta_{p+1}} [R\, \tilde{I}(\theta)^2 + R\, T_p^2 - 2R\, \tilde{I}(\theta)\, T_p]\, d\theta \\ &= E_1 + E_2 + E_3\,, \end{aligned}$$

where $E_1 = \sum_{p=1}^{N} \int_{\theta_p}^{\theta_{p+1}} R\, \tilde{I}(\theta)^2\, d\theta$, $E_2 = \sum_{p=1}^{N} R\, T_p^2 \Delta\theta_p$, and $E_3 = -2\, \sum_{p=1}^{N} \int_{\theta_p}^{\theta_{p+1}} R\, \tilde{I}(\theta)\, T_p\, d\theta$. Moreover, $R = \int_0^\infty \rho\, g(\rho)\, d\rho$, $\tilde{I}(\theta)^2 = \int_0^\infty \rho\, g(\rho)\, I^2(\rho,\theta)\, d\rho/R$, $\tilde{I}(\theta) = \int_0^\infty \rho\, g(\rho)\, I(\rho,\theta)\, d\rho/R$, and $\Delta\theta_p = \theta_{p+1} - \theta_p$. We adopt the "circular" convention that for $p = N$ we have $p+1 = 1$. Note that the integrals $R$, $\tilde{I}(\theta)^2$ and $\tilde{I}(\theta)$ can be precomputed numerically.

To guarantee that the partitions occur where the intensity gradient is high [3] we add the error term

$$E_4 = -\lambda \sum_{p=1}^{N} \int (\frac{\partial \tilde{I}(\theta)}{\partial\theta}|_{\theta_p})^2\, d\theta\,.$$

We can discretize the integral over the angles $\theta$ into a finite sum over of discrete angles $\Delta\theta$. The energy terms $E_i$, $i = 1, 2, 3, 4$ can then be computed numerically as follows. $E_1$ can be precomputed over the whole image and plays no role in the minimization. Then,

$$E_2 = \sum_{p=1}^{N} T_p^2\, R\, [k-j]\Delta\theta \quad \text{and} \quad E_3 = \sum_{p=1}^{N} -2\, T_p\, R \sum_{i=j}^{k} \tilde{I}(\theta_i)\, \Delta\theta\,, \tag{1}$$

where, we remind that $T_p$ is the template value at partition $p$, and partition $p$ starts at angle $\theta_p = j\Delta\theta$ and ends at $\theta_{p+1} = k\Delta\theta$. The last term, $E_4$, is readly computed once the partition angles are chosen.

[1] If the two partitions make an angle of $\pi$ between them, it's an edge.

[2] The angles between the partitions would denote a $Y$, $T$ or other junctions.

[3] The error $E_1 + E_2 + E_3$ do cause partitions to occur on unwanted (due to lack of contrast) places.

## 2.2 The Dynamic Program (DP) Solution

Dynamic programming explores *all* possible solutions to optimize the given form. This is used to minimize $E$. We just need to minimize $[E_2 + E_3 - \lambda E_4]$, to obtain the template values, since $E_1$ is independent of the template and can be precomputed.

The optimization of the error function can be formulated as a dynamic program as shown:

$$C^p_{jk} = \text{cost of fitting } T_p \text{ to region } (j,k), \quad k \geq j$$

If the optimal $I_p$ is obtained then this cost is calculated, and the total error accumulated up to partition $p$ becomes

$$E^p_{sj} = \begin{cases} C^1_{sj} & p = 1, \\ \min_{i<j} \left\{ E^{p-1}_{si} + C^p_{(i+1)j} \right\} & \text{otherwise.} \end{cases}$$

$$E = \min_{s=1,m} E^N_{sM}.$$

$E^n_{sj}$ denotes the error in fitting $n$ lines (or partitions), between angles $\theta_s \ldots \theta_j$. Note that there is a wrap-around involved since $2\pi = 0$. [4] What is left to show is the calculation for $C^p_{jk}$, which requires the calculation for $I_p$. Given a partition, say $p$ between angles $j$ and $k$ , we need to find the optimal $I_p$ and the cost $C^p_{jk}$. Note that the last term, $E_4$ is fixed once the partition has been chosen. Moreover, $E_1$ is independent of the choice of template (though it varies from point to point in the image) thus, we are left to find which $T_p$ minimizes $E_2 + E_3$. From (1) we have

$$\frac{\partial[E_2 + E_3]}{\partial T_p} = 0 \quad \Rightarrow \quad T_p = \frac{\sum_{i=j}^{k} \tilde{I}(\theta_i)}{[k-j]}.$$

Further, substituting back into the formulas, we obtain

$$\min[E_2 + E_3] = -\sum_{p=1}^{N} \frac{R(\sum_{i=j}^{k} \tilde{I}(\theta_i))^2}{[k-j]} \Delta\theta.$$

To compute $C^p_{jk}$ we add the extra term $E_4$. The output of dynamic programing is $N$ pairs of angles and intensities, $T(\{N, (\theta_p, I_p); p \in (1, 2, \ldots, N)\})$.

## 2.3 Blob detection

This approach of fitting piecewise constant features to the image can be expanded to other feature shapes. Suppose we want to find "thik bars" in images. Let us define thik bars on a rectangular window oriented at angle $\theta$. A thik bar is a partition of this window into three stripes and the thik bar represent the middle bar and the other two stripes the background. More precisely, for each orientation $\theta$ we can represent the thik bar with the parameters $\{T_1, T_2, T_3; h\}$, where $T_i$ is the grey value of each stripe and $h$ is half-width of the center stripe (with grey value $T_2$). Again, the mean squared error can be reduced to

[4] If $s > j$, the interval denoted is $\theta_s, \theta_{s+1} \ldots \theta_1 \ldots \theta_j$.

$$\int\int [I(x,y) - T(x,y)]^2 \, dx \, dy = \sum_{p=1}^{3} \int_{h_p}^{h_{p+1}} [\tilde{I}(y)^2 - 2T_p \tilde{I}(y) + T_p^2] \, dy \,,$$

where $h_p$ represent the width of the stripes, and can be derived from $h$ and the width of the window $w$, since $h_1 = \frac{w}{2} - h \;\Rightarrow\; h_2 = \frac{w}{2} + h \;\Rightarrow\; h_3 = w$. An extra term can be added to enforce that the partition occurs at high contrast $\tilde{I}(y)$. The problem of finding the optimal $h$ and the optimal $T_p$ is reduced to the same form as on the junction formulation and can be computed via dynamic programming. One can define a variety of other feature detector in this way. We now study how to use these features to decompose an image for recognition.

# 3 Robust Matching Pursuit

We study the object recognition problem via a robust template decomposition approach. Our main interest is to represent "objects" that appear in a given image with a linear combination of image templates from a well established library of templates. The term "objects" depends solely on the kind of application. For example, if the application is to recognize faces then we should have a template library with a large number of faces (or features of faces). Let the image to be recognized be $I$ and the established template library for some application be $\mathcal{L}$. The task of image recognition is reduced to a function approximation problem of the form

$$I = \sum_j c_j T_j \tag{2}$$

where $T_j \in \mathcal{L}$ (in fact, $T_j$ is a template applied by some translation) and $c_j$ is the choice of coefficients that "best" decompose the image. The difficulty we have is that typically the library $\mathcal{L}$ is large in order to accommodate many possible situations and thus it is an over-redundant basis, i.e., there are infinite many solutions, $c_j$, to this problem.

## 3.1 Brief Review on Matching Pursuit

Inspired by Mallat and Zhang's work [15] we consider a matching pursuit strategy where, at each stage, the criteria of best selection is based on minimizing the image residue. The approach is actually based on a function decomposition method known as *Projection Pursuit* by Friedman and Stuetzle [9]. Projection pursuit is a non-parametric regression method that is concerned with "interesting" projections of high dimensional data and reveals a close connection to the proposed template matching method. Let us be more precise and briefly review this approach.

Suppose we are given a signal $f$, and a library of functions $D = \{g_\gamma\}_{\gamma \in \Gamma}$ where $D$ represents a large, over-redundant family of functions. The *Matching Pursuit* introduced for signal processing in [15] is a greedy stage-wise algorithm and relies on the inner product methods on Hilbert spaces. A "best" (or almost the best) matching library element to the residual signal structures at each stage is decided by successive approximations of the residual signal with orthogonal projections on elements in the library. Thus, at each stage, for any element $g_\gamma \in D$

$$R^{N-1} f = < R^{N-1} f, g_\gamma > g_\gamma + R^N f \tag{3}$$

where $R^N f$ is the $N$-$th$ residue after approximating $R^{N-1} f$ in the direction of $g_\gamma$ (assume that the initial residue is the function $f$, i.e., $R^0 f = f$). The matching pursuit strategy is to find $g_{\gamma^*}$ that minimizes $||R^N f||$ (or the $g_{\gamma^*}$ closest to $R^{N-1} f$), that is

$$||R^{N-1}f - < R^{N-1}f, g_{\gamma^*} > g_{\gamma^*}|| = \min_{\gamma \in \Gamma} ||R^{N-1}f - < R^{N-1}f, g_\gamma > g_\gamma||.$$

We then consider how this approach could be combined with the junction and blob detectors developed in the previous section. We also study how to extend this approach from $L^2$ norms to $L^p$ measures to account for mismatches and occlusions. In this case the difficulty arises, since then, there is no definition of inner product.

## 3.2 A sketch on how to decompose the image into junctions

We here sketch a method on how to decompose the image into multiple junctions. The matching pursuit strategy would first select the features that best fit the image, the ones that give the minimum error, and procede hierarquicaly by removing each template from the image and reconsidering the residual image for the next best fit feature. In the case of features being T-junctions, corners, edges we devise a hierarchical structure where we first find the T-junctions (3 partitions) followed by corners (2 partitions), followed by edges (2 special configured partition, 180 degrees). The intuitive reason for such hierarchie is that at an n-junction (n partitions), one also detects an n-1-junction, thus, the first ones to be detected (and removed) are the higher ones.

At stage $i$ of the pursuit process, when n-junction detection takes place, one finds the best set of parameters $\gamma = \{(x_c, y_c); (\theta_p, T_p) \quad p = 1, ..., n\}$, where $(x_c, y_c)$ is the center of the junction, via the method presented in section 2. Once the parameters are extracted and the detected template removed we can compute the residual image. More precisely, say the detected template is $g_\gamma = \sum_{p=1}^{n} T_p u(\theta, \theta_p, theta_{p+1})$, where $u(x, x1, x2) = 1$, if $x1 \leq x \leq x2$ and zero otherwise. The residual image $R^i f$ becomes

$$R^{i-1}f = g_\gamma + R^i f\,.$$

The difference between this pursuit strategy and the one previously reviewed, is that the choice of template $g_\gamma$ and its coefficient was not based on the inner product $< R^{N-1}f, g_\gamma >$ but rather on the error measure $E_1 + E_2 + E_3 + E_4$ proposed in section 2.

## 3.3 Matching Pursuit with $L^p$ norms

In many applications one may want to look into higher level features, that can not be detected with the methods discussed in section 2. Thus, we now extend the theory of pursuit strategy for a general class of templates, but using a more robust measure than the $L^2$ norm. We choose a cost function based on a concave $L^p$ norm with $0 < \mathbf{p} < 1$. Thus, we don't have a clear definition for inner product on the image space. Nevertheless, the notion of projection can be recaptured via the values of cost function, that is, the criterion for a template to be "best matching" or "closest" to the image is to minimize the values of cost function. In the same way, we have used this to the n-junction decomposition discussed above.

The $L^p$-methods, with $0 < \mathbf{p} < 1$, are often seen in robust regression and function approximation problems [8]. The essence and strengths of $L^p$-methods are originated from their robustness in handling corrupted data and the concavity property of such a norm. There are many local minima to a "concave minimization" problem but, they can be exactly characterized anmd used for an efficient algorithm.

**Library of templates:** Suppose we now have created a well-defined complete template library $\mathcal{L} = \{\tau_j; \quad j = 0, 1, ..., m\} = \{e_0, \tau_1, \tau_2, \cdots, \tau_m\}$ for some application, where $\tau_0 = e_0$ is the canonical template. A canonical template is a trivial template with zero gray-level value pixels everywhere except one pixel at the extreme left and top corner

that its gray-level value is 1. We still have to consider all possible linear transformations $L_i$ for each non-canonical template $\tau_j$. These linear transformations include all possible translations and possibly affine transformations to account for scale and changes of viewing the scene (in this paper, we only study the case that $L_i$ is just a translation). Let the image to be recognized be $I$ of dimension $n$ and each template $\tau_j$ is of dimension $n_T \leq n$ (the dimension of an image is just its total number of pixels). Furthermore, let $P = \{p_1, p_2, \cdots, p_n\}$ and $Q_j = \{q_1, q_2, \cdots, q_{n_T}\}$ be the pixel sets of $I$ and $\tau_j$, respectively. (We order the pixels from top to bottom and left to right.) Let $L_i(\tau_j)$ now represent a translation on template $\tau_j$ such that its first pixel is positioned at the *i-th* pixel $p_i$, i.e. $q1 = p_i$. In order to define a template everywhere in the image and take advantage that in this case $L_i$ is simply a translation we write

$$T_{ij}(p_k) = \begin{cases} \tau_j(p_k - i) & \text{where } 0 \leq p_k - i < n_T, \\ 0, & \text{otherwise} \end{cases}$$

We also define the index set $\Gamma_{ij} = \{p_i, p_{i+1}, ..., p_{(i+n_T}\}$ as the pixels set where the template $T_{ij}$ takes the values from $\tau_j$ . We can rewrite the decomposition equation (2) as

$$I(p_k) = \sum_{i=1}^{n} c_i T_{i0}(p_k) + \sum_{j=1}^{m} \sum_{i=1}^{n} c_\lambda T_{ij}(p_k)$$

where $\lambda = j \times n + i$ and $p_k$ is the *k-th* pixel of the image.

Let us consider the matching pursuit strategy. Let the initial residual image $R^0 I = I$. Then, at stage $N$, if template $T_{ij}$ is chosen, the *N-th* residual image can be derived as follow:

$$R^N I(p_k) = R^{N-1} I(p_k) - c_\lambda T_{ij}(p_k) \qquad \text{for } k = 1...n \quad \text{and} \quad \lambda = j \times n + i.$$

Thus, $R^N I$ can be derived by "projecting" $R^{N-1} I$ in the direction of $T_{ij}$ and the best matching $T_{ij}$ is the one that minimizes the residue at stage $N$.

**Cost function ($L^p$-method)** The cost function $F$, at stage $N$, using the $L^p$ norm is defined as

$$F_p(c_\lambda) = \frac{1}{|c_\lambda|^p \cdot |Variance(\tau_j)|^{\frac{p}{2}}} \sum_{k \in \Gamma_{ij}} |r_k|^p = \sum_{k \in \Gamma_{ij}} |\tilde{r}_k|^p \tag{4}$$

where $r_k = |R^{N-1} I(p_k) - c_\lambda T_{ij}(p_k)|$ is the *k-th* residue at pixel $p_k$, $k \in \Gamma_{ij}$ when $R^{N-1} I$ is matched by the template $T_{ij}$. The normalized residues

$$\tilde{r}_k = \frac{r_k}{|c_\lambda| \cdot |Variance(\tau_j)|^{\frac{1}{2}}}$$

are used to avoid "over-utilization" of templates on darker regions of the image.

**Local Minima:** Due to the concavity of the cost function $F_p(c)$, the solutions to the reduced linear system that are local minima to $F_p(c)$ are at the vertices of the polytope that is the intersection between the subspace expanded by the reduced linear system and the $n_T$ hyperplanes, each defined by $c_k = 0$, $k \in \Gamma_{ij}$ (see for example [7]). For each hyperplane $c_k = 0$ the solution of the reduced linear system becomes $c_k = I(p_k)/T_{ij}(p_k)$. This results guarantee that we only need to search for $n_T$ coefficients $c_k$ to obtain the one that minimizes $\tilde{r}_k$. Typically, the size of templates are not large so the estimation of $c_k$ and $\tilde{r}_k$ can be rather efficient.

# 4 Shape feature representation and decomposition

It is interesting to also consider features that capture global properties of the objects, such as "shape features". Following Cootes and others [5,16] we represent the shapes using the principal components of the shapes over the training set. Our training set is derived from the profiles from different views of the same object and we will discuss its implications.

Our shape representation is based on the method of Principal Components (also called principal modes, the modes of variation, or the Karhounen-Loeve transform). Consider a collection of two-dimensional shapes $(x_1, x_2, ...x_m)$ each embedded in an image. Here we think each shape as being a profile of an object taken from different views. More precisely, each shape $x_i = (x_{i1}, y_{i1}, ...x_{in}, y_{in})$ is defined by $n$ points so that there is a natural correspondence between the points for any two shapes $x_j$ and $x_k$ (i.e., the corresponding points define some landmark in each of the images). Considering the training set of an object, by taking multiple profiles from different views, we compute the mean shape as the average of all shapes as

$$\bar{x}_k = \frac{1}{m}\sum_{i=1}^{m} x_{ik}\,.$$

We then compute the covariance matrix C

$$\mathbf{C}_{lk} = \frac{1}{m}\sum_{i=1}^{m}(x_{il} - \bar{x}_{il})(x_{ik} - \bar{x}_{ik})\,.$$

Finally, we compute the eigenvectors and corresponding eigenvalues of C so that

$$\mathbf{C} = \mathbf{PSP^{-1}}\,,$$

where $\mathbf{S}$ is the diagonal matrix of eigenvalues and the columns of $\mathbf{P}$ are the unit eigenvectors of $\mathbf{C}$. The matrix $\mathbf{P}$ defines the modes of variation of the shapes, and the variance associated with each mode is expressed in the associated eigenvalue. We order the columns of $\mathbf{P}$ and $\mathbf{S}$ so that $S_{ii} \geq S_{i+1,i+1}, \quad \forall i < 2n$. Then, $\mathbf{P}$ defines a new set of orthogonal basis vectors for our set of shapes. The modal representation for a shape $x$ (whether or not it is part of our training set) is then defined by the equation

$$x = \bar{x} + \mathbf{P}b\,,$$

where $b$ is a vector of weights for the $2n$ modes. Each column of $\mathbf{P}$ is comprised of $n$ two-dimensional displacement vectors, one for each point, so that by varying the weights one can create new instances of the family of shapes. The $i'th$ column of $\mathbf{P}$ is referred to as the $i'th$ modal shape.

The modal representation provides a statistically orthogonal basis, so that to the largest degree possible, the modes represent independent shape features. This basis is ordered so that the low-order modes provide coarse features with the detail enhanced by the higher-order modes. This basis is optimal for the description of the training set in the sense that for any $n < mag(b)$, the sum of the squared errors between the training set and its approximation as a linear combination of the first $n$ modes is minimal over all possible orthogonal bases [13,17].

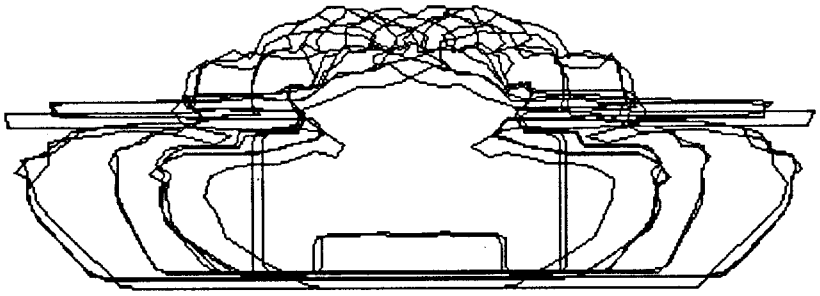

Figure 1: Contours from different views normalized and located at the gravity center.

Figure 2: The First three modes, a., b., c., where the display represents the possible displacements with respect to the mean.

## 4.1 Setting up the Training Set

To build the training set, we take projections of a three-dimensional model of the vehicle of interest from many different points of view. For purposes of illustration we show variation only over the rotation angle, though in practice our training sets are constructed over a range of rotation, depression, and tilt angles. We store each of these profiles tagged with the parameters of the view.

In order to compute the modal representation, we need a correspondence between the members of the training set. The correspondence is important because the modes will model variance over the locations of the corresponding points of the profiles. We use the following simple algorithm to extract equally spaced points along the profiles. We first center the profile at the center of gravity. We then select points $(X, Y_{low})$ and $(X, Y_{high})$ along the curve such that the length of the curve (in the counter-clockwise direction) from $(X, Y_{low})$ to $(X, Y_{high})$ is the same as the length of the curve from $(X, Y_{high})$ to $(X, Y_{low})$. We sample evenly around the curve between these points. This process produces a list of $n$ points $(X_0, Y_0, X_1, Y_1, ...X_{n-1}, Y_{n-1})$ for each profile, where $X_0 = X, Y_0 = Y_{low}, X_{n/2} = X, Y_{n/2} = Y_{high}$, and the rest of the points are evenly spaced between these two. This method produces nicely correlated points because the points $(X_0, Y_0)$ and $(X_{n/2}, Y_{n/2})$ move only up and down (ie. the modal shapes for the training set have $X$ component equal to zero for these two points); while the rest of the points scale with the length of the profile, which tends to vary smoothly over the training set.

In Figure 1 we illustrate some of the correlated contours used to build the training set. Figures 2a–c show the first three modes of variance extracted from the training set, each varied from plus 10 percent to minus 10 percent of the total variance of the mode over the training set.

## 4.2 Image shape-feature decomposition

Our approach is similar to the methods of deformable contours, or snakes [3,12], where an energy function has two terms, one for the image features and one for the shape. Our improvements are as follow. First, we search for the minimal energy shape configuration in the lower-dimensional space of the modes rather than in the space of the shapes. Second, we make use of the hiearchical ordering of the modes and multi-resolution image features to allow for a coarse-to-fine search in both the image feature space and the shape feature space. Finally, we provide a framework in which these methods can be used for images with occlusions, and for shapes which are not normally distributed about the mean. We use a training set constructed from profiles of military vehicles from many points of view, and

apply these methods to the identification of these vehicles in sensor images.

**The energy function:** In our formulation the energy function assigned to each shape is based solely on how well the image features at the current configuration match the expected image features. This is because, different than on our recent investigations using medical images, the assumption of normal distributions does not hold for profiles of military vehicles. Since our goal is to find the maximal probability configuration of the shape in the image, we define the energy in a manner analogous to the Mahalanobis distance:

$$E = -\ln(\Pr(x_i)),$$

where $x_i = \{x_{i1}, x_{i2}, ...x_{in}\}$ is a configuration of the shape. To compute $\Pr(x_i)$, we consider each component of $x_i$ independently. In the studies we have done, the $x_{ik}$ are either the pixel values, the values of the image gradients, or the directions of the image gradients, at the control points of the shape. The probability of the shape occurring given the image evidence (assuming all configurations of the shape are equally likely) is

$$\Pr(x_i) = \prod_k \Pr(x_{ik}).$$

In the case where a portion of the shape is obstructed from view, the image features $x_{ik}$ at the control points are considered to be random. For the unobscured $x_{ik}$, we expect a relatively small Gaussian variance which is derived from the expected variance in the pixel values. Hence, the probability distribution for each xi is

$$Prob(x_{ik}) = P_0 + (1 - P_0)e^{-(x_{ik} - \bar{x}_{ik})/2\lambda^2},$$

where $P_0$ is the probability that this point is obscured from view and $\lambda$ is the variance of $x_{ik}$ about it's mean, $\bar{x}_{ik}$.

For the studies described, we use the direction of the gradients in the image so our energy term is a function of the difference between the actual and expected directions of the gradients in the image. We choose the direction of the gradients regardless of magnitude because they are relatively stable under various sensor conditions and so are well suited to the identification of solid structures in range images.

At full resolution, the expected gradients are simply the perpendicular vectors to the curve. At lower resolutions, however, the expected gradient directions may change because finer shape features may be obscured by more gross features. To determine the expected directions of the gradients for each profile in the training set, we compute multiresolution images from binary representations of the profiles and store the gradients along the profiles at each level of resolution.

## 4.3 The Search Algorithm

Given the modal representation which captures the variance of the profiles over the training set, we construct a mapping from the real-world parameters (angles of rotation) to the modal parameters for each profile. This will allow us to perform our search in the modal shape feature space while constraining our search to the profiles found in the training set. In practice, this mapping is stored in a matrix indexed by the real-world parameters, so that each entry contains the modal weights. More precisely, we compute the vector of modal weights

$$b = \mathbf{P}^{-1}(X - \bar{X}),$$

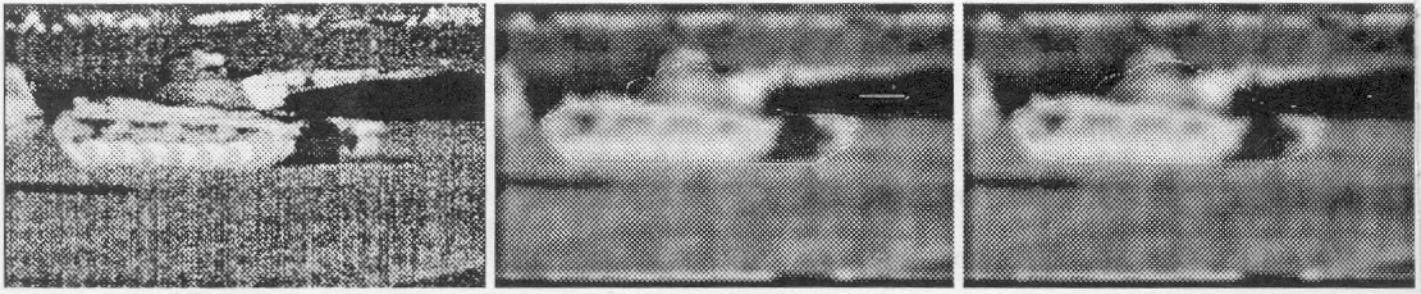

Figure 3: a. Original FLIR image enhanced with histogram equalization. b.The lowest energy segmentation c. A reduced resolution version of the resulting contour approximated using only the first three modes.

needed to reconstruct each profile and store these at the entries of the matrix which corresponds to the parameters of this view of the object.

The exhaustive approach is to try each point in sequence, selecting the parameters which yield the lowest energy, or highest probability. This is not practical for a large data set with more than one degree of freedom. We have implemented a method which uses multi-resolution image analysis and the hierarchical ordering of shape features provided by the modal shape representation. The idea is that reduced resolutions allow gross changes in the shape using the large-variance modes, while the fine adjustments need to be made only at high image resolutions. The algorithm is outlined below.

$possible - hits :=$ all locations in the image (quantized by the lowest level of resolution)

Loop for each level of image resolution, from low to high

Determine the relevant subset of modes

Loop for each location in $possible - hits$, until convergence

Adjust each modal weight until it is locally minimal

$possible - hits :=$ all those locations whose energy is less than a fixed threshold.

The relevant subset of modes is that set which moves some points more than one pixel distance in the current image resolution. Similarly, the adjustment for each mode is that step which move some points in the modal shape more than one pixel distance.

Our search is constrained to the set of shapes found in the training set, so that adjusting each mode moves us to a new entry in our **S** matrix - and hence provides a new value for all of the modal weights. Since the modes comprise a linearly independent orthogonal basis for the set of shapes, we expect (and have observed) that when a single mode reaches a non-global minima, it will be shifted out of the minima by a strong gradients in the energy for other modes.

This method has been successfully applied to sensor images. The number of configurations tested is orders of magnitude fewer than the exhaustive approach, even with the threshold used for further consideration of locations set quite high. We illustrate the results of one segmentation. The original image, an FLIR image enhanced with histogram equalization for purposes of display is shown in Figure 3a. We illustrate the lowest energy segmentation in Figure 3b. Figure 3c shows a reduced resolution version of the resulting contour approximated using only the first three modes (the modes shown in Figures 2a–c) overlaid on the Gaussian filtered image. At the displayed level of resolution, only the first three modes were used for the search, because only these modes had components greater than the effective pixel size at this level of resolution.

## References


[1] J. Ben-Arie and K. R. Rao, "On The Recognition of Occluded Shapes and Generic Faces Using Multiple-Template Expansion Matching", Proceedings IEEE International Conference on Pattern Recognition, New York City, 1993

[2] Cohen, I., Ayache, N., Sulger, P. "Tracking Points on Deformable Objects Using Curvature Information," In Proceedings of the Second European Conference on Computer Vision 1992, Santa Margherita Ligure, Italy (1992) pp. 458-466.

[3] Cohen, I., Cohen, L.D., Ayache, N. "Using Deformable Surfaces to Segment 3D Images and Infer Differential Structures," Computer Vision, Graphics, and Image Processing: Image Understanding, Vol. 56, No. 2, (September 1992) pp. 242-263.

[4] Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J. "Training Models of Shape from Sets of Examples," Proceedings of the British Machine Vision Conference, Springer-Verlag, 1992, pp. 9-18.

[5] Cootes, T.F., Taylor, C.J., Lanitis, A., Cooper, D.H., Graham, J. "Building and Using Flexible Models Incorporating Gray-Level Information," Proceedings of the Fourth International Conference on Computer Vision, Berlin Germany (May 1993) pp. 242-246.

[6] R. Deriche, T. Blaska, "Recovering and Characterizing image features using an efficient model based approach," CVPR (1993) pp.530-535

[7] M. J. Donahue and D. Geiger, "Template Matching and Function Decomposition Using Non-Minimal Spanning Sets", Manuscript, 1994.

[8] H. Ekblom, "$L_p$-methods For Robust Regression", *BIT* 14, p.22-32, 1973.

[9] J. H. Friedman and W. Stuetzle, "Projection Pursuit Regression", *Journal of the American Statistical Association*, vol. 76, p.817-823, 1981.

[10] A. Guzman. Computer recognition of three dimensional objects in a visual scene. Technical Report MAC-TR-59, MIT, 1968.

[11] D.A. Huffman. Impossible objects as nonsense senteces. In E. B. Meltzer and D. Michie, editors, *Machine Intelligence*, Vol.6. Edinburgh Univ. Press, Edinburgh, U. K., 1971.

[12] Kass, M., Witkin, A., and Terzopolous, D. "Snakes: Active Contour Models," International Journal of Computer Vision, Vol. 1 (1987) pp. 321-331.

[13] Kirby, M., Sirovich, L., "Application of the Karhunen-Love Procedure for the Characterization of Human Faces." IEEE Transactions on Pattern Analysys and Machine Intelligence, Vol. 12, No. 1, (January 1990).

[14] Y. Lamdan, J. T. Schwartz and H. J. Wolfson, *"Geometric Hashing: A General and Efficient Model-Based Recognition Scheme"*, Proc. ICCV2, p.238-249, Dec. 1988.

[15] S. Mallat and Z. Zhang, "Matching Pursuit with Time-Frequency Dictionaries", *IEEE Trans. on Signal Processing*, Dec. 1993.

[16] Pentland, A., Sclaroff, S. "Closed-Form Solutions for Physically Based Shape Modeling and Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 4 (July 1991).

[17] Sirovich, L., Kirby, M. "Low-Dimensional Procedure for the Characterization of Human Faces." *Journal of the Optical Society of America*, Vol. 4, No. 3, pp. 519- 524 (March 1987).