

VIRTUAL SAMPLE GENERATION FOR TEMPLATE-BASED SHAPE MATCHING

D.M. Gavrilu and J. Giebel

Image Understanding Systems, DaimlerChrysler Research, Ulm 89081, Germany
{darius.gavrilu,jan.giebel}@DaimlerChrysler.com, www.gavrilu.net

ABSTRACT

This paper presents a method for improving the performance of matching systems that correlate using shape templates. The basic idea involves extending an existing set of training shapes with generated "virtual" shapes, in order to improve representational capability. Yet no a-priori feature correspondence is necessary among the original shapes in the training set. Instead, an integrated clustering and registration approach partitions the original shape samples into clusters of similar and registered shapes; in each cluster a separate feature space is embedded. This allows for each cluster the derivation of standard compact parameterizations. This paper demonstrates that sampling these low-order spaces can produce an extended training set which facilitates a superior matching performance, as measured by a ROC curve. In the experiments, we consider a realistic application involving thousands of pedestrian shapes and perform correlation matching based on distance transforms.

1. INTRODUCTION

For many interesting object detection tasks there are no explicit prior models available to support a matching process. This is typically the case for the detection of complex non-rigid objects under unrestricted viewpoints and/or under changing illumination conditions. This paper deals with shape-based systems which capture object appearance by a set of shape templates and which use correlation as basic tool for matching. The appeal of such systems lies in their robustness and generality. Their robustness is derived from a template-driven approach which reduces the burden on error-prone segmentation. Correlation copes relatively well with missing data due to occlusion or incorrect segmentation. Their generality relates to the ease in dealing with a broad class of shapes, without requiring feature correspondences or parameterizations of the training shapes.

Among appearance-based systems using shape templates, those that correlate using distance transforms [1] have proven to be particularly successful. The smoothness of the resulting correlation measure with respect to shape perturbations and changes in transformation parameters enables the use of very efficient pruning and hierarchical techniques for matching [3] [12] [6].

An important prerequisite, however, for such appearance-based systems is that their training set adequately covers the object's shape distribution. All possible object instances should ideally have a dissimilarity value below the matching threshold with respect to existing templates. Yet the ability to collect training samples is for many applications limited by practical reasons. This paper presents a technique for enlarging a set of training shapes with generated, "virtual" shapes, for improving representational capability (for the use of virtual samples outside the shape domain, see e.g. [15]). The examples used throughout this paper involve closed contours, although extensions to open contours are possible. The method is general in the sense that it does not require prior feature correspondence among the shapes in the training set.

Our approach is summarized in Figure 1. From an original set of training shapes, we derive K separate parameterizations for the shape distribution, based on an integrated clustering and registration approach, followed by dimensionality reduction. In each of the K resulting feature spaces we fit and sample a Gaussian distribution to obtain virtual samples for an extended training set. The main result of this paper is that the proposed shape parameterization and re-sampling method can result in an extended training set which facilitates a superior matching performance, as measured by a ROC curve.

The outline of this paper is as follows. The next Section reviews previous work on shape learning. Our approach to shape modeling and sampling is presented in Section 3. Section 4 provides the experiments, followed by the conclusions in Section 5.

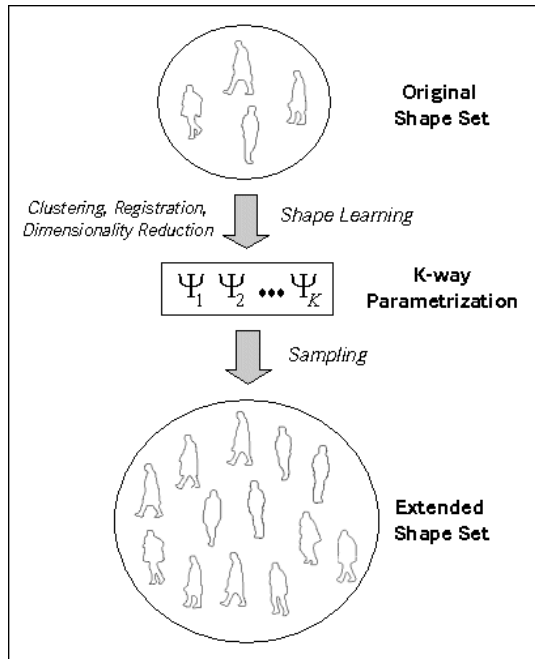


Figure 1: Extending the "physical" training set with "virtual" samples

2. PREVIOUS WORK

A fair amount of literature exists on learning shape models from examples. A typical approach involves two components: a method for shape registration and a method for deriving feature spaces and associated parameterizations.

Shape registration methods bring shapes into correspondence and normalize for certain geometrical transformations. They follow a similar succession of steps: shape decomposition, feature selection, point correspondence and finally, alignment. The first step, shape decomposition, involves determining control ("landmark") points along a contour and breaking the shape into corresponding segments. This can be done by considering the extrema of the curvature function [13] [16], or by considering a criticality measure based on the area of three successive points [18].

The next step involves selecting features that are transformation invariant (e.g. translation, rotation and scale); these features are derived from straight-line (i.e. polygonal) [5] [7] [11] [13] or Fourier approximations [9] of the shape segments. Similarity measures are typically based on length ratios and angles between successive segments for the polygonal case (e.g. [13]) and weighted Euclidean metrics on the low-order Fourier coefficients, for the Fourier approximations (e.g. [9]

[17]).

At this point, correspondence between the control points of two shapes can be established by means of either combinatoric approaches [5] [13] or by sequential pattern matching techniques such as dynamic programming [7] [9]. The latter has the advantage of inherently enforcing ordering constraints and being efficient, while at the same time, producing an optimal solution. After correspondence between control points has been established (and possibly, between interpolated points), alignment with respect to similarity transformation is achieved by a least-squares fit [4].

Registration establishes point correspondence and aligns a pair of shapes. The straightforward way to account for N shapes is to embed all N shapes in a single feature vector space, based on the x and y locations of their corresponding points. This is done either by selecting one reference shape (typically, the "closest" to the others) and aligning all others to it, or by employing a somewhat more complex hierarchical procedure [11]. The resulting vector space allows the computation of various compact representations for the shape distribution, based on radial (mean-variance) [8] or modal (linear subspace) [2] [4] [11] decompositions. Combinations are also possible [10].

Our approach, discussed in the next Section, differs from previous work by the use of an enhanced shape registration method based on multiple shape scales [14] and the use of a more general shape representation based on multiple feature spaces (similar to [5]). Yet arguably the most important distinction concerns the general procedure. Rather to derive shape parameterizations from examples and use them directly for tracking purposes, we proceed with a re-sampling step which reverts samples back to their template representations. We do this because we are primarily interested in the object detection application and would like to take advantage of the before-mentioned efficient template systems based on distance transforms (now equipped with an improved training set).

3. SHAPE LEARNING AND (RE)SAMPLING

To establish a sound basis for shape learning, our first step in Figure 1, we compared four shape registration methods. These were obtained by considering all possible combinations between before-mentioned two techniques for control point detection (Gaussian-based filtering of the curvature function [13] and critical point detection [18]) and two techniques for feature selection

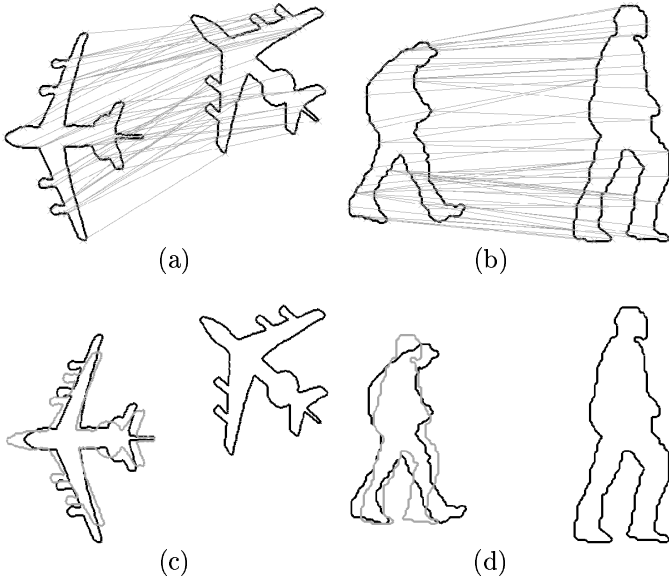


Figure 2: Established point correspondences between two (a) planes and (b) pedestrians and their pairwise alignment, (c) and (d). Aligned shapes are superimposed in grey.

and similarity measurement (using polygonal approximations [13] and piecewise Fourier decompositions [9]), respectively. We chose as matching algorithm invariably dynamic programming, because of its efficiency and optimality property, see previous Section.

Extending previous shape registration methods, we represented shapes at multiple scales. For the method of [13], this is achieved by using multiple Gaussian σ values (e.g. [14]), whereas in [18] this involves using different area criteria. For registration, we selected among all scales the one which minimized the mean alignment error. Our study involving several large datasets (e.g. pedestrians, planes) consistently identified the hybrid based on Gaussian curvature filtering for control point detection [13] and Fourier coefficients features [9], as best performing; this hybrid was used to embed shapes in feature spaces, as described below. See Figure 2 for some registration results under similarity transform; the registration method is quite successful here in pairing up physically corresponding points (e.g. the tip and wings of the planes, the heads and feet of the pedestrians).

It is, however, an open secret that existing automatic shape registration do not always produce the nice results of Figure 2. In fact, they tend not to be able to deal well with appreciably different shapes. For example, even with our extension to multi-scale shape representation, none of the shape registration

variants we analyzed were able to correctly register a pedestrian shape with the two feet apart with one with the feet together. Approaches that have initially all shapes embedded in a single feature space (i.e. the result of a registration step having established point correspondences across the entire training set) [2] [4] [11] typically involve benevolent data sets, incorporate application-specific constraints for registration, or allow for some manual processing.

We opted instead for a more general registration approach, which does not forcibly try to embed all N shapes into the same feature vector space. Instead, it combines shape clustering and registration, embedding only the (similar) shapes within a cluster into the same vector spaces. The idea is similar to [5], but makes use of a different clustering algorithm. Rather than computing all possible $N \times N$ shape registrations off-line and iteratively selecting a reference shape to which to map all the remaining matching shapes to, like in [5], we incrementally register our shape templates to various existing prototype, and update the latter at each iteration. Our clustering algorithm has a K -means-like flavor:

0. pick an initial shape S_1 and add it to cluster C_1 as prototype:
 $C_1 = \{S_1\}, P_1 = S_1$
 - while** there are shapes left do
 1. select one of remaining shapes: S_k
 2. compute mean alignment error $d(S_k, P_i)$ from element S_k to the existing prototypes P_i , where i ranges over the number of clusters created so far
 3. Compute $d_{min} = d(S_k, P_j) = \min_i d(S_k, P_i)$.
if $d(S_k, P_j) > \theta$
then assign S_k to a new cluster C_{n+1} :
 $C_{n+1} = \{S_k\}, P_{n+1} = S_k$
else assign S_k to existing cluster
 $C_j = \{S_{j1}, \dots, S_{jn}\}$ and update its prototype:
 $C_j = C_j \cup \{S_k\}$
 $P_j = \text{Mean}(S_{j1}, \dots, S_{jn}, S_k)$
- end**

Integrated in the clustering algorithm, at Step 2, is our best performing hybrid shape registration method for establishing point correspondences and computing alignment errors. The resulting point correspondences are used for the prototype computation in Step 3, the latter being the Procrustes average [8]. Parameter θ is a user-defined dissimilarity threshold that controls the number of clusters to be created. Figure 3 illustrates some typical clustering results.

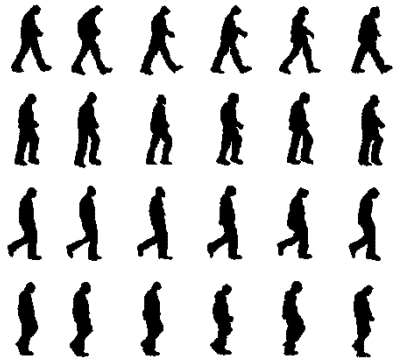


Figure 3: Shape clustering: each row contains the elements of a different cluster

The output of the registration approach is a set of K clusters containing similar shapes for which point correspondences have been established. These enable the introduction of K separate feature space by considering the x - and y - coordinates of (corresponding) shape points within a cluster. Subsequently, a principal component analysis is performed on each of the K feature spaces to obtain a more compact representation. The dimensionality of the reduced feature spaces varies, depending on how many dimensions are needed to meet a user-supplied variance criterion (e.g. capturing 95% of variance).

The last step in Figure 1 involves the generation of new samples; this is achieved by fitting in each of the K reduced feature spaces a Gaussian distribution and sampling it. Samples are then transformed back to their original template representation.

4. EXPERIMENTS

Generating virtual samples is not beneficial by itself. The purpose of the experiments is to demonstrate that an original set of training shapes can be extended in a way that it produces concrete advantages in terms of matching performance, offsetting inevitable drawbacks related to increased matching effort and higher memory requirements.

To quantify these advantages, we considered a real-world application involving pedestrian matching. The application is of interest in the context of smart vehicles, where by detecting dangerous situations ahead of time, one aims to reduce (or avoid altogether) the effects of a collision. Earlier work showed the promise of a template matching system based on distance transforms (e.g. [6]). Therefore, we opted for correlation

with distance transforms as basic matching tool for our experiments, and in particular for the chamfer distance [1]. Matching using chamfer distance in its most basic form is illustrated in Figure 4. It involves two binary images, a template (Figure 4b) and an edge image (Figure 4c) derived from the original scene image (Figure 4a). Matching consists of applying the chamfer distance transform on the edge image, resulting in a distance image (Figure 4d) which contains at each pixel an approximation of the Euclidean distance to the nearest feature pixel in the corresponding edge image. See Figure 4d, increasing distances are shown in lighter shades of grey. The template is then positioned over the distance image, and one considers the pixel values of the distance image which lie beneath the feature pixels of the template. The lower these distances are, the better the match. One typically averages the individual distance contributions to obtain an overall measure of match.

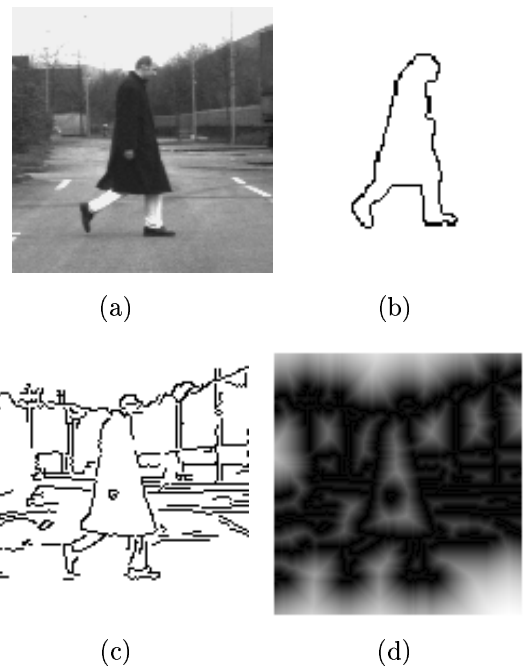


Figure 4: (a) original image (b) template (c) edge image (d) DT image

Our data set contained roughly 5000 pedestrian shape templates, extracted from real images in a quite time-consuming manual labeling process. The data set was subdivided in a training and test set of about 2500 shape templates each, with no overlap between the two sets. The training set consisted only of the pedestrian templates. The test set contained, in addition, a total of 5000 noise or "garbage" templates, cropped image edge structures which did not contain pedestrians.

These garbage template were selected from those image parts which showed already a good chamfer distance match with respect to the pedestrian templates (i.e. were not sampled randomly).

The original training set of 2500 shapes was extended using the methods described in the previous Section. Re-sampling was done in a manner that increased the original training size by a factor of 6. The newly generated virtual samples replaced, rather than augmented, the original samples in the extended training set. Two cases are now considered, the test set against the original training set, and the test set against the extended training set.

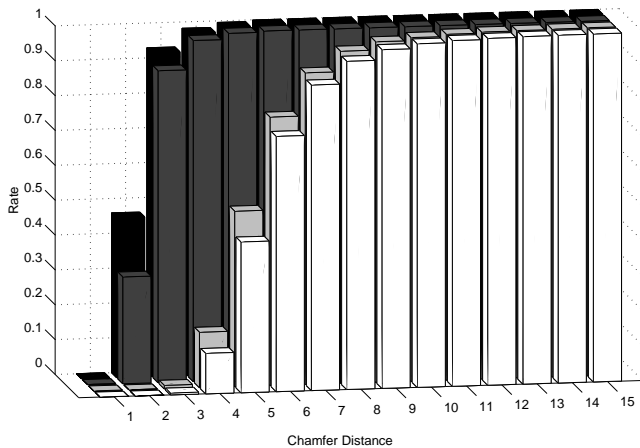


Figure 5: Cumulative histogram of chamfer distances: original pedestrian vs. pedestrian (dark grey), original pedestrian vs. garbage (white), extended pedestrian vs. pedestrian (black), extended pedestrian vs. garbage (light grey). First and second term refer to object classes in training and test set, respectively.

Figure 5 shows the four relevant cumulative histograms of chamfer distances between the object classes in the training and test set, for the original and extended training set. The underlying distance distributions were computed by looping through the elements of the relevant object class in the test set and aggregating the minimum chamfer distances (i.e. the "best" match) with respect to the pedestrian templates in the training set. From Figure 5 one observes that, the pre-selection notwithstanding, elements of the garbage class are more dissimilar to those of the pedestrian class, than the elements of the pedestrian class are among themselves (see dark grey versus white histograms, respectively). Furthermore, the distances from both the pedestrian and garbage test set to the training set decrease, when extending the latter. See Figure 6 for the (virtual)

pedestrian-garbage template pairs with the lowest chamfer dissimilarity measure.

From the two cumulative histograms involving the original (or extended) training set, one derives the corresponding ROC curve, the rate of correct detections versus false positives. This is done by considering a particular chamfer distance threshold (x-axis of histogram) and identifying the fraction of pedestrian and garbage samples which have lower distance values (the detection and false positive rate, respectively). The resulting two curves are shown in Figure 7. The figure quite convincingly demonstrates the benefit of our virtual shape generation procedure; one observes that in a sizable (and application-relevant) detection rate interval (0.75 - 0.90), the ROC involving the extended training set outperforms the one related to the original training set. For this interval, the false positive rate is reduced by at least factor 1.5 by equal detection rates. But Figure 7 also illustrates another important point, that by increasing the detection rate above a certain threshold (here at about 0.93) one no longer profits from a given virtual sample set. At a certain point, the associated chamfer distance thresholds become so large that any representational "gaps" in the pedestrian distribution are already covered by the original training samples. Increasing the distance thresholds further will now mainly increase the false positive rate. From Figure 7 one observes that this turning point occurs at a high false positive rate exceeding 0.015 for the current virtual sample set. On the other hand, one could try to sample the object shape distribution even more finely, to attempt to "squeeze" yet more performance out of the ROC curve.

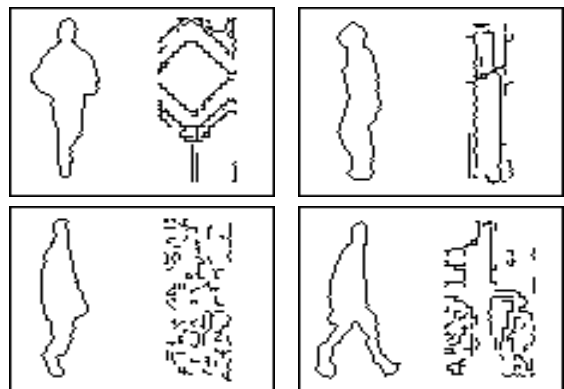


Figure 6: The most similar (virtual) pedestrian-garbage pairs in the data set

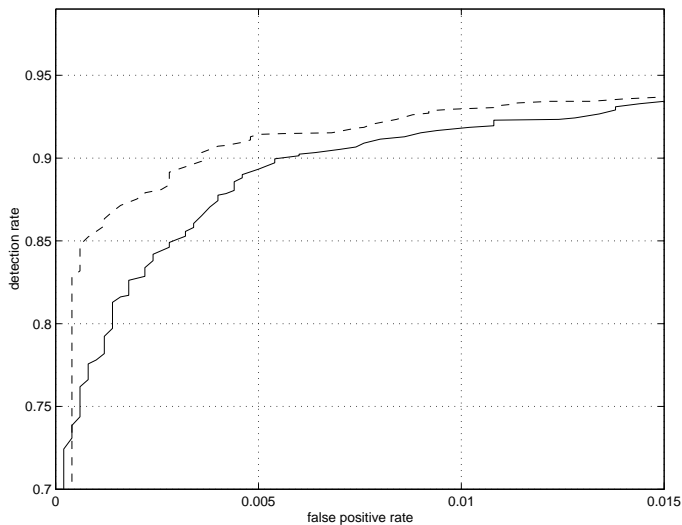


Figure 7: ROC performance for the original training set (solid line) and the extended training set (dotted line)

5. CONCLUSIONS

This paper presented a general method for artificially enlarging a set of N shapes, without requiring a-priori shape correspondence. The method involved a parameterization-re-sampling framework, in which training shapes were first partitioned into K groups, by combined shape registration and clustering. Thereafter, point correspondences were used to derive reduced feature spaces which then were sampled. We demonstrated that this approach can be beneficial for a template-based matching system in terms of improving ROC performance.

6. REFERENCES

- [1] H. Barrow et al. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *International Joint Conference on Artificial Intelligence*, pages 659–663, 1977.
- [2] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In *European Conference on Computer Vision*, pages 299–308, 1994.
- [3] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1988.
- [4] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and applications. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
- [5] N. Duta, A. Jain, and M.-P. Dubuisson-Jolly. Learning 2d shape models. In *International Conference on Computer Vision*, pages 8–14, Kerkyra, Greece, 1999.
- [6] D. M. Gavrila and V. Philomin. Real-time object detection for “smart” vehicles. In *International Conference on Computer Vision*, pages 87–93, Kerkyra, Greece, 1999.
- [7] Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1312–1328, December 1999.
- [8] C. Goodall. Procrustes methods in the statistical analysis of shape. *J. Royal Statistical Soc. B*, 53(2):285–339, 1991.
- [9] J. Gorman, R. Mitchell, and F. Kuhl. Partial shape recognition using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):257–266, 1988.
- [10] T. Heap and D. Hogg. Improving the specificity in PDMs using a hierarchical approach. In *Proc. of the British Machine Vision Conference*, Colchester, UK, 1997. BMVA Press.
- [11] A. Hill, C. Taylor, and A. Brett. A framework for automatic landmark identification using a new method of nonrigid correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):241–251, March 2000.
- [12] D. Huttenlocher, G. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.
- [13] H.-C. Liu and M. Srinath. Partial shape classification using contour matching in distance transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(11):1072–1079, November 1990.
- [14] F. Mokhtarian and A. K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992.
- [15] D. A. Pomerleau. *Neural Network Perception for Mobile Robot Guidance*. Kluwer Academic Publishers, Boston, 1993.
- [16] A. Rattarangsi and R. T. Chin. Scale-based detection of corners of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4):430–449, 1992.
- [17] L.H. Staib and J.S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1061–1075, 1992.
- [18] P. Zhu and P. Chirlian. On critical point detection of digital shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):737–748, 1995.