Learning Multiscale Image Models Of 2D Object Classes

Benoit Perrin, Narendra Ahuja and Narayan Srinivasa
The Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
405 N. Mathews Avenue, Urbana, IL 61801

Abstract

This paper is concerned with learning the canonical gray scale structure of the images of a class of objects. Structure is defined in terms of the geometry and layout of salient image regions that characterize the given views of the objects. The use of such structure based learning of object appearence is motivated by the relative stability of image structure over intensity values. A multiscale segmentation tree description is automatically extracted for all sample images which are then matched to construct a single canonical representative which serves as the model of the class. Different images are selected as prototypes, and each prototype tree is refined to best match the rest of the class. The model tree for the class is that tree which is best supported over all the initializations with different prototypes. Matching is formulated as a problem of finding the best mapping from regions of example images to those of the model tree, and implemented as a problem in incremental refinement of the model tree using a learning approach. Experiments are reported on a face image database. The results demonstrate that a reasonable model of facial geometry and topology is learnt which includes prominent facial features.

1 Introduction

This paper is concerned with learning the canonical gray scale structure of the images of a class of objects. Structure is defined in terms of the geometry, layout and photometric relationships among salient image regions that are common across given object images. The use of such structure based learning of object appearance is motivated by the relative stability of image structure over intensity values across multiple views. Since region boundaries in image often correspond to scene discontinuities in illumination, albedo, objects, etc., these boundaries are more invariant to changes in illumination level and viewpoint than the intensity values themselves. Thus, any descriptions of objects inferred in terms of the region structures are expected to be fairly stable representations of object images.

The region structure of an image may be represented by a tree that captures the different geometric and photometric scales, and geometric and topological interrelationships characterizing the image regions. Large regions are said to have a coarse spatial scale while smaller sizes are said to be associated with finer spatial scales. Given a tree that represents such a multiscale segmentation of an image, the goal is to extract a tree that serves as the common denominator of the trees corresponding to all object images. We use an algorithm that automatically extracts the tree representation of an image without any *a priori* knowledge of the scales present. The trees derived from different object images are matched to construct a canonical representation or model that provides maximal degree of match to all the trees. The model tree for the class is derived as that tree which is best supported over all the initializations. The matching of each prototype tree to the rest of the class is formulated as a problem of finding the best matches from regions of example images to those of the model tree, and implemented as a problem in incremental learning. The learning is realized using a modified Fuzzy Adaptive Resonance Thoery (Fuzzy ART) architecture that incrementally refines a prototype tree based on the features of the matched regions of example images.

2 Previous Work

There have been previous efforts at finding canonical aspects of the images from a single class. However, most of these are limited to relatively unstable features such as the responses of an edge detector or, edge segments. Shams [15] presents an approach to applying canonical graph representation for pattern recognition. He uses local features of orientation and intensity edges for recognizing objects such a plane or a tank. Von der Marlsburg et al. [17] uses an architecture for pattern recognition based on graph matching to recognize faces [8] and to achieve invariance with respect to some location and orientation variabilities [7]. To improve upon the unreliability of the isolated local features such as used in these methods, [5] uses simple homogeneity criteria to split and merge tiles defined by an *a priori* tessellation method to obtain more stable segmentations, and thus more meaningful regions. However, *a priori* chosen, low level criteria do not yield meaningful candidate regions for canonical class descriptions from real images which is the objective of this paper. Methods have been developed that integrate *a priori* domain knowledge into the segmentation which improves segmentation at the expense of making the methods domain specific [19, 16, 18, 12, 3, 6].

Moghaddam and Pentland [9] present a probabilistic method for 2D object detection. This method requires the *a priori* identification of important object features. It has been applied to faces and hands images, gives interesting results even with few eigenspaces, and seems robust with respect to noise and data variations. Murase and Nayar in [11] describe a 3D method based on eigenspace generation. Instead of using intrinsic shape information, which is often difficult to extract due to lighting variations, they match appearance. As in [9], a suitable number of eigenspaces for each object must be found. Hornegger and Niemann in [4] propose a statistical framework for learning, localizations and identifications of objects. One of the most prominent connectionist methods is based on the radial-basis function (RBF) network introduced by Poggio and Girosi [14] to learn a mapping from an input space to an output space. It was applied by Poggio and Edelman [13] to recognize three-dimensional stick figures from two-dimensional images. Mukherjee and Nayar in [10] use a learning algorithm to obtain the network parameters using basis functions generated from wavelet decomposition of different training images of objects. The work described in this paper is aimed at obtaining a canonical representation or model of the object images such that the model is explicitly defined in terms of relatively low level, and thus domain independent, features, which are also relatively independent of imaging conditions.

3 Overview of Learning Approach

Each image is converted into a segmentation tree [1] in which the levels are indexed by a photometric scale parameter. Initially, the prototype is stored in the form of the segmentation tree of one of the sample object images. The segmentation trees of several other samples are then presented to the learning system one image at a time. For every sample image presented, each of its regions at the coarsest scale is a candidate for matching with the prototype regions at the same scale. The set of all single region candidates constitutes a *unary* hypothesis set. In addition to single region candidates, additional regions are also generated as candidates by merging two and three neighboring regions in the region adjacency graph (RAG) of the sample image segmentation tree giving rise to *binary* and *ternary* hypothesis sets, respectively. The union of unary, binary and ternary hypothesis sets of a sample image constitutes the hypothesis set for that sample image. A coarsest-scale hypothesis set consisting of unary, binary and ternary hypotheses is analogously generated for the prototype. These sample and prototype hypotheses sets are paired to provide the best matches of region features such as area, centroid and shape (measured as the eccentricity of the best fitting ellipse). The matching is performed using a learning algorithm. At the finer scales, the hypothesis sets are generated, for both the prototype and the sample image, from regions that are the children and/or neighbors of matched regions at the next coarser level. While these hypothesis sets also contain unary, binary or ternary intrascale regions, the generation of these regions is guided by the matching process at a coarser scale. This prevents the creation of irrelevant hypothesis sets at finer scales.

The prototype hypothesis sets that find matches with each sample image hypothesis set, at each scale, are modified in order to reflect the features in the sample image that are different from the prototype. In addition to the modifications to the prototype features, the learning algorithm also stores the frequency with which each region of the prototype hypothesis set is matched with the regions belonging to the

sample image hypothesis set of the training samples. Matched prototype regions with the highest frequencies indicate the regions that are most salient. The final result of the learning process is the set of regions that is most salient based on all training samples starting with a given prototype. While this approach is stable with respect to different prototype initializations, we use the commonly extracted regions across several initializations as the canonical representation for the object class. This removes any bias due to specific prototype initializations.

4 Choice of Features for Region Matching

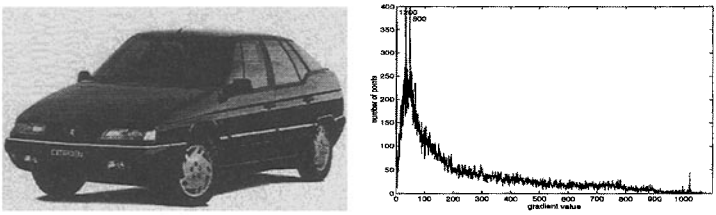
Several features are used to determine the quality of match between a pair of regions. The work in this paper focusses on learning canonical models of facial image structure. These images contain a face in the center and there is little background area. All the sample images are normalized to the prototype image such that each face image appears in the center and has about the same size. Region centroids represent their relative positions, and are used as one type of feature. Region area is used as another feature. The eccentricity of the region is used as its shape feature, and is approximated by that of its best-fitting ellipse.

In order to derive these three normalized features, the two-dimensional moment of order $(p+q)$, for an NXM discretized image $g(x,y)$ is defined as $m_{pq} = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} x^p y^q g(x,y)$. The centroid (c_x, c_y) of each region is defined as $c_x = m_{10}/m_{00}$ and $c_y = m_{01}/m_{00}$ where m_{00} is the area of the region. This area is computed as the number of pixels within each region. To compute the eccentricity, we first compute the central moments μ_{pq} of the region by replacing x and y in the expression for m_{pq} by expressions $(x - c_x)$ and $(x - c_y)$. These moments are then normalized with respect to scale as $\nu_{pq} = \mu_{pq}/m_{00}^{\frac{p+q}{2}}$. The normalized eccentricity η, which gives a shape measure invariant to translation and scale, is computed as $\eta = \sqrt{(\nu_{20} - \nu_{02})^2 + 4\nu_{11}^2}/m_{00}$.

At the coarsest scale only, the area and centroid features were used. The eccentricity information was not used at this scale because, the regions at the coarsest scale are normally large. These large regions are not very stable in it shape feature. By traversing to finer scales from the coarse level region, its features take a more definite shape. For example, a face region at the coarsest scale, extracted as a single region or by merging two or three regions, is usually not well defined. However, at finer scales, the regions that are found within it can correpsond to the eye or mouth and these are far more well defined in their shape. Thus, at the coarse scale, only the area and centroid were used as features for the matching process. At finer scales, the area, centroid and its eccentricity were used.

5 Criteria for Merging Regions

The merging of regions is an important and necessary step in our approach. This is because lighting effects lead to smooth shading in images. During the multiscale segmentation, this may further lead to splitting of shaded regions with subregions at arbitrary locations, and thus to the creation of spurious regions. To alleviate this problem, methods need to be developed to merge erroneous subregions into the original parent region. Since there is no sharp change in gray-level value across the border between spurious regions, the gray-level gradient across the border will be low. This property is used to detect mergable regions. If the gradient at most of the border pixels has a shallow slope, then the two regions can be merged. The exact condition for merging depends upon the definition of the terms *most of the border pixels* and the *degree of shallow slope*. For this purpose, we use two thresholds: T_{per} (*most of the border pixels* means more than T_{per}%) and T_{gr} (*degree of shallow slope* means slope smaller than T_{gr}). To compute these thresholds, the histogram of gradient values at the border pixels is plotted. An example of the histogram for the image of a car is shown in Figure 1. Experiments suggest that T_{gr} is best if located at the tail end of the steep part of curve, just before it flattens. The threshold T_{per} for the number of border points is selected such that T_{gr} is greater than T_{per}. In this paper, T_{per} was set typically between 80 and 90%. Thus, two regions R_1 and R_2 can be merged if $|B| * 100/|N| > T_{per}$ where B is a border point. A point on the R_1-R_2 region boundary is a border point if $G(B) < T_{gr}$ where $G(B)$ is gradient at the border point. N is the total number of border points between regions R_1 and R_2. The term $|x|$ here refers to the sum of the number of elements in x. An important consequence of reducing these spurious regions is the reduction in the number of regions that have to be merged to



(a) (b)

Figure 1: (a) Original picture of the car on which experiments were based. (b) Histogram of the gradient values on border points for the car.

generate the hypothesis sets. This increases the efficiency of the matching and learning process. The three features that define each region have to be re-computed after merging of two or more regions occurs. They are computed from the parent regions unless the regions have already been merged in other training samples and their values are available.

6 Coarse-To-Fine Generation of Hypothesis Sets

At the finest scale, the segmented image typically contains several small regions. Finding the canonical representation at this scale would require the generation of several mergings. Therefore, it is desirable to use the knowledge that comes from a matching at a coarser scale to guide the matching at a finer scale. Using the segmented tree, the algorithm searches for matching regions among the children of the regions that have been matched at the previous coarser scale level. This confines the search to interesting areas, thus limiting the number of hypotheses. Because the coarsest scale has fewer regions than the other scales, it can be used to initialize the process without expensive computations.

At each scale, the hypothesis sets are generated for the sample and prototype images in two steps. First, two sets of unary regions or *unary hypotheses* are extracted from the segmented images at the current scale. One is extracted from the prototype RAG, and another from the current sample RAG. To select the unary hypotheses at the coarsest scale level, all the regions of RAG are selected for matching. Below this level, hypotheses are selected in two ways. Let us assume that a match has been found between prototype and sample RAG. Then for the finer scales, children of these matched regions are selected to compose a pair of unary hypothesis set. Neighbors of these children are also added to the sets. This process is repeated for every matched region of the preceeding coarse level.

For the regions in the prototype that are not children of matched regions, a centroid-based hypothesis generation strategy is used, where the region that has a nearby centroid in the sample image is selected. Neighbors of these two regions are also selected to form two sets of unary hypotheses. The centroid-based method generates small but numerous sets of hypotheses. However, experiments have shown that, in most cases, the number of matches obtained from using the first strategy is more important than from the centroid-based method. Therefore, the first strategy can still be used for objects or scenes where the positioning of regions is not precise by possibly expanding the neighborhood. Once the pair of unary hypothesis sets is formed, unary hypotheses are merged to generate binary and ternary hypotheses. The output of the hypothesis generation process is, therefore, two sets of regions which are candidates for matching.

7 Learning Algorithm and Architecture

In this approach, a learning architecture adapted from Fuzzy ART algorithm [2] was chosen to perform matching because it is self-organizing, robust to noise, and massively parallel, which makes it useful for on-line pattern recognition applications. One of the most important properties of the architecture used

is that the majority of processing involves simple compare and add operations, resulting in an efficient learning algorithm.

7.1 Fuzzy ART Representation and Notation

In developing an algorithm to perform the above computation, we have used fuzzy set theory [20] to represent classes as well as perform computations [2]. For concreteness, we will explain the notation for the 2-dimensional space; it generalizes to other spaces in a straight forward manner.

Class: A class is represented by specifying two diagonally opposite vertices of its rectangle. This is done by a vector consisting of the coordinates of one vertex followed by the complement (with respect to 1) of the coordinates of the diagonally opposite vertex. Thus, for example, the output class represented by the rectangle defined by vertices (x_1, y_1) and (x_2, y_2) is represented by the vector $(x_1, y_1, 1 - x_1, 1 - y_1)$. For a (class consisting of) a single point (vertex) (x_1, y_1), the representation is the 4-tuple $(x_1, y_1, 1 - x_1, 1 - y_1)$, denoted by its *weight* vector $\mathbf{W}$.

Norm: The norm $|V|$ of a vector (class) is the sum of the city block distances of the class from the points $(0, 0)$ and $(1, 1)$.

Distance: The distance between a sample point and a class rectangle is denoted by the city block distance to the nearest point in the class.

AND Operation: The fuzzy AND (or $\wedge$) between two classes is the vector whose elements are obtained by taking pairwise MIN of the corresponding elements of the operand vectors. AND of two classes (points) denotes the result of adding one to the other, possibly requiring expansion.

Choice Function: The choice function is used to determine the class defined by its weight $\mathbf{W}$ that is closest to a given point. Given a new point $\mathbf{I}$ and a class $\mathbf{W}$, the choice function is defined as $\frac{|\mathbf{I} \wedge \mathbf{W}|}{|\mathbf{W}|}$, which assumes highest value for that class which is at shortest distance from $\mathbf{I}$. If the choice function is one for a given class, then the class is a *fuzzy subset choice* for input $\mathbf{I}$. This means that the input $\mathbf{I}$ is completely contained within the class $\mathbf{W}$. If more than one category is a fuzzy subset choice, then a small but positive parameter α is added to the denominator to break the tie such that the class that maximizes $|\mathbf{W}|$ among the fuzzy subset choices is chosen.

Vigilance function: The vigilance function is used to enforce the restriction on class size. For example, given a sample $\mathbf{I}$ and a class $\mathbf{W}$, $\mathbf{W}$ is allowed to (expand and) include $\mathbf{I}$ if the value of the vigilance function, defined as $\frac{|\mathbf{I} \wedge \mathbf{W}|}{|\mathbf{I}|}$, is no smaller than a certain *a priori* (user specified) threshold ρ called the vigilance parameter.

7.2 The Modified Fuzzy ART Algorithm

The modified Fuzzy ART algorithm consists of two layers as shown in Figure 2. The F_1 layer is called the *input* layer while the F_2 layer is called the *prototype* layer. The F_1 layer receives each region m belonging to the sample image hypothesis set as an input vector $\mathbf{S_m}$. This vector is defined as: $\mathbf{S_m} = (S_{m1}, S_{m2}, \cdots, S_{mM})$ where the parameters S_{mi} (for $i = 1, \cdots M$), in general, represent the characteristic features of each region such as area, centroid, color, etc.

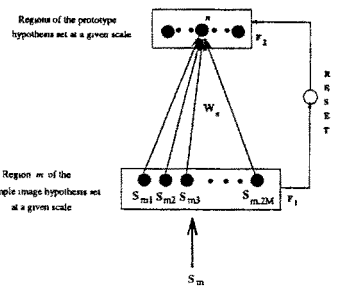


Figure 2: The modified fuzzy ART network.

The learning algorithm is outlined now. First, initialize the components of the weight $\mathbf{W}_n$ corresponding to each region n of the prototype hypothesis set to the M-dimensional feature value vector $\mathbf{P}_n = (P_{n1}, P_{n2}, \cdots, P_{nM})$. Select a region m from the sample image hypothesis set as input. Then, present the feature vector $\mathbf{S}_m = (S_{1m}, S_{2m}, \cdots, S_{Mm})$ to F_1. This input and its complement are stored as a single vector $\mathbf{I}$ in F_1. Compute the class $n \in \mathbf{P}_n$ that is closest to the input $\mathbf{I}$ using the choice function T_n as $T_n = \frac{|\mathbf{I} \wedge \mathbf{W}_n|}{|\mathbf{W}_n|}$ Form the hypothesis that the selected class n is the appropriate classification for the given input. Then, test the hypothesis using the vigilance criterion as $\rho_n = \frac{|\mathbf{I} \wedge \mathbf{W}_n|}{|\mathbf{I}|} \geq \rho$ where ρ is the vigilance parameter set by the user.

If class n satisfies the vigilance criterion, then the input $\mathbf{I}$ is added to the list L_{mn}. If there are more regions in the sample image hypothesis set, then repeat the above steps for these additional regions. When all the regions are processed, determine the maximum value in the list L_{mn} for each region n of the prototype hypothesis set. For each such region, update the weights W_n using the features of the matched region m as $\mathbf{W}_n = \beta|\mathbf{I} \wedge \mathbf{W}_n| + (1-\beta)\mathbf{W}_n$ where β controls the rate at which the features of the matched input region m are allowed to refine the weight $\mathbf{W}_n$. Typically, $\beta \leq 0.2$ during training so that only the regions that have extremely good matches are allowed to refine the weights $\mathbf{W}_n$. The bound on the size of the hyperrectangle, $|D_n|$, for each class n can be defined as $|D_n| \leq M(1-\rho)$ where M is the number of features in the input. Thus, if the vigilance parameter ρ is small, the size of the hyperrectangles are bigger and vice versa. The training process continues until the input feature space is covered with hyperrectangles. The frequency of winning for all winning nodes in F_2 is incremented by 1. If there are more training samples, then repeat the above steps for the new samples. It should be noted that there is a separate Fuzzy ART network for each scale. At the end of the learning process, the regions with the highest frequency of winning are the regions that correspond to the 2D model for the given training samples.

8 Experiments and Results

The proposed algorithm was tested for 20 different initializations of face images (from the O.R.L. face database) with all initializations being frontal views in a neutral position. For each initialization, 400 training images of 92x112 each were used to train the network. These images were presented in different orders. The segmentation transform used four different scales. The scales are ranked from level 0 to 3 with the coarsest scale corresponding to level 0 and the finest scale to level 3. The feature vector for matching consisted of area and centroid information for all scales. The canonical representations have been found to be most stable with the following range of fuzzy ART parameters, from scale 0 to 3 where $\alpha = (0.85, 0.9, 0.9, 0.9)$ and $\beta = (0.1, 0.2, 0.1, 0.2)$. This range of parameters allows the algorithm to converge slowly to canonical features.

8.1 Results for Different Initializations

To illustrate the effect of different initializations, consider the original gray level images of two different faces as shown in Figure 3(e) and Figure 4(e). These images are segmented into quite different initial regions by the multiscale transform as shown in columns (a) and (d) of these figures. The coarse scale segmentation on top of the column provides a good estimate of the quality of initialization: while the image in Figure 4 has many features, such as the eyes or mouth in the right place, it is more difficult to find these features for the image in Figure 3. Despite these different initializations, the algorithm extracts similar features after being trained on 400 different face images. In column (b) and (f) the most frequently found regions are shown. These regions are approximated with ellipses and superimposed on a face image not used in the learning process called the *neutral face*. The area, centroid and eccentricity for each region is extracted from the weight vector as the mean of the weight vector and of its complement corresponding to each region. In these figures, the 8 best regions of the coarsest scale level (level 1), 10 best regions for level 2 and 3 and 15 best regions for the two last pictures (scale level 4 without and with eccentricity information) are shown. This selection was based on whether the regions were matched at least for half of the training examples presented. Furthermore, it has been observed that the extracted features are similar among all initializations for this choice of number of regions at each scale. Columns (c) and (g) show the number of times each pixel of the image has been matched by the 25 most frequently found regions. Here, the darker the pixel, the more frequently it

has been matched. Because the silhouette of the face is often found at the coarsest level as can be seen in columns (c), the images for the coarsest level are much darker i.e., more regions are matched. The most frequently matched regions show that the algorithm concentrates on the area with the obvious features: eyes, mouth.

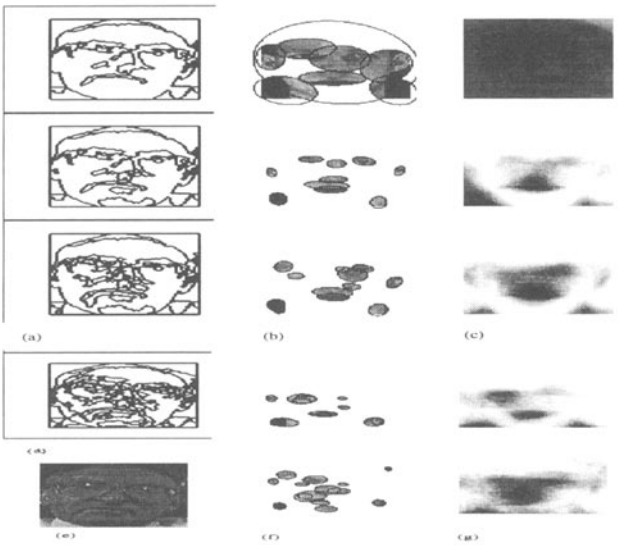


Figure 3: For scale levels 0 (row 1) to 2 (row 3): (a) Original segmentation tree. (b) Ellipse approximation. (c) Pixel match frequencies. (d) Original segmented image for scale level 3. (e) Original gray-scale image. (f) Ellipse approximation for scale level 3 without using shape for matching (top), using shape for matching (bottom). (g) Pixel match frequencies without shape (top), with shape (bottom).

8.2 Results after Postprocessing

As mentioned in Section 3, the results obtained from different initializations are used to select the most commonly found regions across all these initializations. Results of this postprocessing is illustrated using Figure 5. The regions represents the final canonical representation at each scale level. This representation was computed using the most frequently occurring regions (ranging from 8 for scale 0 to 15 for scale 3) across all initializations. The Figures (b) are the elliptic approximations of the most frequently matched regions. Match frequencies from the postprocessing show a clear difference between the matched and unmatched regions. This enables us to extract the most frequent regions. For each of these stable regions, the pixels that have been most frequently matched are represented in columns (c). The darker pixels correspond to the more frequently matched regions. At finer scale levels, there exists a clear threshold to separate the most frequent regions from others. The effect of thresholding (60 % percent of the maximum frequency) the stable regions in columns (c) is shown in columns (d). By superimposing the thresholded regions in columns (d) into a single image, we obtain the result shown in columns (a). Despite the variabilities in the database and with a limited range of possible matching regions (only up to ternary hypothesis sets), these results show that the algorithm has been able to efficiently extract the main features of faces.

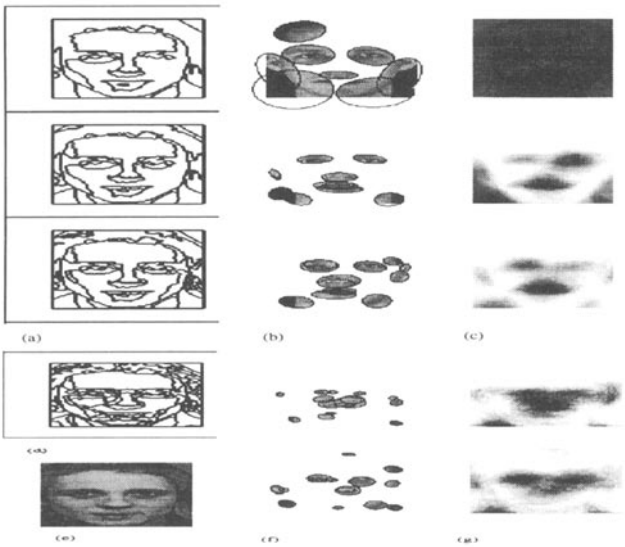


Figure 4: For scale levels 0 (row 1) to 2 (row 3): (a) Original segmentation tree. (b) Ellipse approximation.(c) Pixel match frequencies. (d) Original segmented image for scale level 3. (e) Original gray-scale image. (f) Ellipse approximation for scale level 3 without using shape for matching (top), using shape for matching (bottom). (g) Pixel match frequencies without shape (top), with shape (bottom).

References

[1] N. Ahuja. A transform for multiscale image segmentation of integrated edge and region detection. pages 1211–1235, 1996.

[2] G. A. Carpenter, S. Grossberg, and D. B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[3] J. R. Beveridge et al. Segmenting images using localized histograms and region merging. *International Journal of Computer Vision*, 2(3):311–347, 1989.

[4] J. Hornegger and H. Niemann. Statistical learning, localization and identification of objects. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 914–919, 1995.

[5] S. L. Horowitz and T. Pavlidis. Picture segmentation by a directed split-and-merge procedure. In *Proc. International Conference on Pattern Recognition*, pages 424–433, 1974.

[6] I. Y. kim and H. S. Yang. A systematic way for region-based image segmentation based on markov random field model. *Pattern Recognition Letters*, (15):969–976, 1994.

[7] W. K. Konen, T. Maurer, and C. von der Malsburg. A fast dynamic link matching algorithm for invariant pattern recognition. *Neural Networks*, 7(6):1019–1030, 1994.

[8] M. Lades, J. C. Vorbr'uggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. W'urtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–310, 1993.

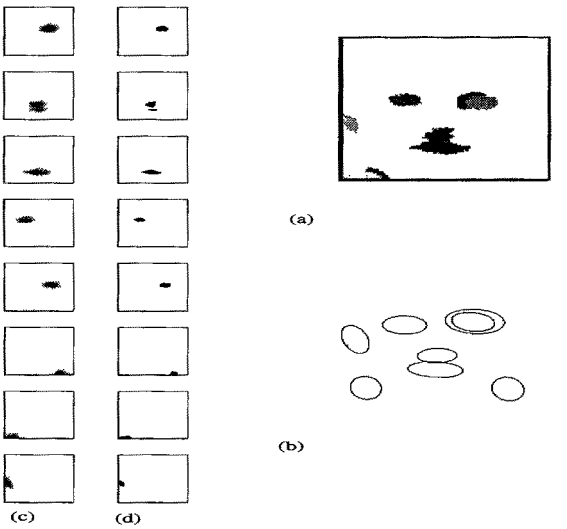


Figure 5: Scale level 1: (a) Shape of the stable regions. (b) Ellipse approximation of the stable regions.(c) Pixel match frequencies for each detected feature.(d) Result of the thresholding (c).

[9] B. Mogghaddam and A. Pentland. Probabilistic visual learning for object detection. In *Proc. IEEE International Conference on Computer Vision*, pages 786–793, 1995.

[10] S. Mukherjee and S. K. Nayar. Automatic generation of grbf networks fo visual learning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 794–800, 1995.

[11] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, (14):5–24, 1995.

[12] A. M. Nazif and M. D. Levine. Low level image segmentation:an expert system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(5):555–577, 1984.

[13] T. Poggio and S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343:236–266, 1990.

[14] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78:1481–1497, 1990.

[15] S. Shams. Multiple elastic modules for visual pattern recognition. *Neural Networks*, 8(9):1439–1456, 1995.

[16] J.M. Tenenbaum and H. G. Barrow. Experiments in interpretation-guided segmentation. *Artificial Intelligence*, 8:241–274, 1977.

[17] C. von der Malsburg and E. Bienenstock. A neural network for the retrieval of superimposed connection patterns. *Europhysics Letters*, 3(11):1243–1249, 1987.

[18] D. I. Waltz. Generating semantic descriptions from drawings of scenes with shadows. Technical Report A. I. Memo 1271, M. I. T. Artificial Intelligence Laboratory, 1972.

[19] Y. Yakimovsky and J. A. Feldman. A semantics-based decision theory region analysis. In *Proc. International Joint Conference on Artificial Intelligence*, pages 580–588, 1973.

[20] L. Zadeh. Fuzzy sets. *Information Control*, 8:338–353, 1965.