RuCCS TR-36
August, 1997

# View-Based Object Recognition Using Saliency Maps

Ali Shokoufandeh
Center for Cognitive Science and
Department of Computer Science
Rutgers University
shokoufa@cs.rutgers.edu

Ivan Marsic
Department of Electrical and
Computer Engineering
Rutgers University
marsic@caip.rutgers.edu

Sven Dickinson
Center for Cognitive Science and
Department of Computer Science
Rutgers University
sven@cs.rutgers.edu

Technical Report #36
Center for Cognitive Science (RuCCS)
Rutgers University, Busch Campus
Psychology Building Addition
New Brunswick, NJ 08903

# View-Based Object Recognition Using Saliency Maps*

Ali Shokoufandeh[§] Ivan Marsic[†] and Sven J. Dickinson[§]

[§]Department of Computer Science and
Center for Cognitive Science
Rutgers University
New Brunswick, NJ 08903
{shokoufa,sven}@cs.rutgers.edu

[†]Department of Electrical and
Computer Engineering
Rutgers University
New Brunswick, NJ 08903
marsic@caip.rutgers.edu

**Abstract**

We introduce a novel view-based object representation, called the *saliency map graph (SMG)*, which captures the salient regions of an object view at multiple scales using a wavelet transform. This compact representation is highly invariant to translation, rotation (image and depth), and scaling, and offers the locality of representation required for occluded object recognition. To compare two saliency map graphs, we introduce two graph similarity algorithms. The first computes the topological similarity between two SMG's, providing a coarse-level matching of two graphs. The second computes the geometrical similarity between two SMG's, providing a fine-level matching of two graphs. We test and compare these two algorithms on a large database of model object views.

**Keywords:** View-Based Object Recognition, Shape Representation and Recovery, Graph Matching.

*A condensed version of this paper will appear in the IEEE International Conference on Computer Vision, Bombay, January 1998, under the title, "View-Based Object Matching."

# 1 Introduction

The view-based approach to 3-D object recognition represents an object as a collection of 2-D views, sometimes called aspects or characteristic views [14]. The advantage of such an approach is that it avoids having to construct a 3-D model of an object as well as having to make 3-D inferences from 2-D features. Many approaches to view-based modeling represent each view as a collection of extracted features, such as extracted line segments, curves, corners, line groups, regions, or surfaces (Ikeuchi and Kanade [11], Burns and Kitchen [2], Ullman and Basri [30], Dickinson et al. [7], and Pope and Lowe [21]). The success of these view-based recognition systems depends on the extent to which they can extract their requisite features. With real images of real objects in unconstrained environments, the extraction of such features can be both time consuming and unreliable.

In contrast to the feature-based view-based recognition paradigm, a number of image-based view-based recognition systems have emerged. Beginning with the eigenface approach proposed by Turk and Pentland [29], these image-based approaches avoid extracting complex features from an image; instead, they retain the entire raw image as a single feature in a high-dimensional space. Turk and Pentland focused on the domain of faces and therefore did not require a large set of model views for each face. Nayar and Murase extended this work to general 3-D objects where a dense set of views was acquired for each object [19].

Although avoiding costly and often unreliable feature extraction, these image-based approaches pay the price of sensitivity to lighting conditions, image translation, image rotation, depth rotation, occlusion, and minor shape variation, all of which affect an image's pixel values and result in a change in the image's location in some high-dimensional space. Recent results have shown some progress towards solving these problems, e.g., the work of Belheumer and Kriegman [1] (limited invariance to illumination changes) and the work of Leonardis and Bischoff [15] and Schmid and Mohr [24] (limited invariance to occlusion). Nevertheless, the lack of abstraction from raw image data to the model means that the model defines a very specific object instance.

The concept of computing coarse-to-fine image descriptions has much support in the

computer vision community; some examples include [3, 12, 16, 20, 27, 28]. In some cases, attention models have been developed that use a multiscale description to decide where in the image to apply some operation. Lindeberg has based this selection process on a quantitative analysis of gray-level blobs in scale space [16]. Jägersand [12] uses an information theoretic measure to compute "informativeness" of image regions at different scales, while others have defined some measure of "importance" and used it to drive an attention process [20, 27, 28]. Although suitable for locating objects in images for further processing, the above multiscale descriptions, often called saliency maps, lose the detailed shape information required for object recognition.

Some multiscale image descriptions have been used to locate a particular target object in the image. For example, Rao et al. use correlation to compare a multiscale saliency map of the target object with a multiscale saliency map of the image in order to fixate on the object [23]. Although these approaches are effective in finding a target in the image, they, like any template-based approach, do not scale to large object databases. Their bottom-up descriptions of the image are not only global, offering little means for segmenting an image into objects or parts, but offer little invariance to occlusion, object deformation, and other transformations.

An approach similar to the approach we will present is due to Crowley et al. [5, 4, 6]. From a Laplacian pyramid computed on an image, peaks and ridges at each scale are detected as local maxima. The peaks are then linked together to form a tree structure, from which a set of *peaks paths* are extracted, corresponding to the branches of the tree. During matching, correspondence between low-resolution peak paths in the model and the image are used to solve for the pose of the model with respect to the image. Given this initial pose, a greedy matching algorithm descends down the tree, pairing higher-resolution peak paths from the image and the model. Using a log likelihood similarity measure on peak paths, the best corresponding paths through the two trees is found. The similarity of the image and model trees is based on a very weak approximation of the trees' topology and geometry, restricted, in fact, to a single path through the tree.

In this paper, we present a multiscale view-based representation of 3-D objects that, on

one hand, avoids the need for complex feature extraction, such as lines, curves, or regions, while on the other hand, provides the locality of representation necessary to support occluded object recognition as well as invariance to minor changes in both illumination and shape. In computing a representation for a 2-D image (whether model image or image to be recognized), a multiscale wavelet transform is applied to the image, resulting in a hierarchical saliency map of the image that offers advantages over a Laplacian pyramid. This saliency map is represented as a hierarchical graph structure, called the *saliency map graph*, that encodes both the topological and geometrical information found in the saliency map.

The similarity between a test image and a model image is defined as the similarity between their respective saliency map graphs. We address the problem of matching two saliency map graphs, leading to two matching algorithms. The first algorithm finds the best mapping between two saliency map graphs in terms of their topological structure, while the second algorithm factors in the geometry of the two graphs. In each case, we present an evaluation function that determines the overall quality of the match, i.e., the similarity of the two graphs. We demonstrate and evaluate our image representation and our two matching algorithms using the Columbia University COIL image database. In addition, we assess the viewpoint invariance of our representation and matching algorithms.

# 2 A Scale-Space Saliency Representation of an Image

To reduce the complexity in matching input image representations to model view representations, we seek a scale-space or coarse-to-fine representation of images that allows us to first match or index based on the coarse-level features in the image. Coarse-level correspondence can then be used to constrain a fine-level matching of the remaining features. Furthermore, we would like our image representation to be invariant to slight variations in the illumination falling on the object, image-plane rotation, translation, and scaling of the object, slight rotation in depth of the object, slight deformations of the shape of the object, e.g., stretching, bending, etc., and occlusion of the object.

Traditional view-based object representations that are image based, e.g., [29, 19, 1, 15,

24, 23], are neither coarse-to-fine nor invariant to the above transformations due to the global nature of their representations (although some offer limited invariance to particular transformations). However, the advantage of these approaches is that complex feature extraction, grouping, or abstraction is not required. Systems based on more invariant view-based image descriptions, e.g., [11, 2, 30, 7, 21], have relied on complex feature extraction (e.g., edges, lines, regions, etc.) which is not only unreliable but often requires domain-specific parameter tuning.

To address these shortcomings, we compute a scale-space representation of an image in which image objects (homogeneous regions) are located at the coarsest scale which captures their salient shape properties. Moreover, both the geometrical and topological relations between the regions will be explicitly encoded in the representation. Finally, computing these regions and relations requires the setting of very few parameters.

## 2.1 The Multiscale Wavelet Transform

The scale-space image representation that we have selected is based on a multiscale wavelet transform [25]. The advantage of the wavelet decomposition lies in its effective time (space)–frequency (scale) localization. Unlike other image transforms, e.g., [3, 4], which spread the information across their basis functions, the wavelet transform allows us to compute better localized object representations. In the output of the transform, as illustrated in Figure 1, the salient shape of small objects is best captured by small wavelets, while the converse is true for large objects. Searching from finer to coarser scales (right to left in Figure 1), we select the scale which captures the most efficient encoding of an object's salient shape; above the chosen scale, extraneous information is encoded, while below the chosen scale, the object is overly "blurred." The region defining the object at the chosen scale is called the *scale-space cell* (SSC) [18].

The dyadic wavelet transform of a function $f \in L^2(\mathbb{R})$ at the scale $2^j$ and at the position $k$ is given by the inner products of the function with the family of wavelets

$$(Wf)(j,k) = \langle f, \psi_{j,k} \rangle = 2^{-j/2} \int_{-\infty}^{+\infty} f(x) \overline{\psi(2^{-j}x - k)}\, dx \qquad (1)$$

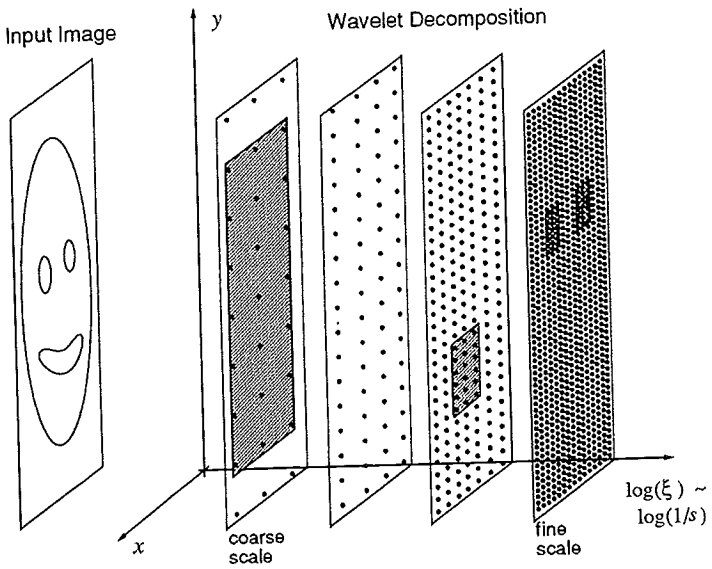



Figure 1: In the proposed multiscale description, objects are captured at the coarsest scale which captures their salient properties. Note that the frequency $\xi$ and the scale $s$ are inversely proportional.

where the overline denotes the complex conjugate. This inner product may also be viewed as a convolution product $(Wf)\ (j,k) = f * \overline{\psi_{j,k}}$ or as a filtering of the function $f$ with a band-pass filter whose impulse response is $\overline{\psi_{j,k}}$.

Detecting the scale-space cells requires analysis of the wavelet transform response at each scale. The scale-space cell of an image object is located at the scale which is approximately one octave below the scale at which the object's response becomes indistinguishable from other image objects of the same size. At this scale, the object's response resembles the wavelet basis function impulse response. The following subsections will explore scale-space cell detection in greater detail.

## 2.2 Scale-Space Cells in One Dimension

To illustrate the detection of scale-space cells, consider the one-dimensional signal in Figure 2, using the wavelets described in [17]. Any object behaves like a point at all scales coarser than its characteristic scale $I$. This means that the wavelet transform of a signal at any

scale $j > I$ is the same no matter what the object's shape is; the transform is completely determined by the width of the objects and their amplitude. We can ignore the information at the scales $j > I$ and still be able to reconstruct the original signal almost perfectly, as illustrated in the right column in Figure 2.

The reconstructed signal $f_6^r(x)$ is obtained by replacing the responses $(W\cdot)(6,x)$ of objects A and B with the analyzing wavelet $\psi_{6,k}$ of the same amplitude. Similarly, $f_{6,5}^r(x)$, $f_{6,5,4}^r(x)$, and $f_{6,5,4,3}^r(x)$ are obtained by successively replacing the corresponding wavelet responses at the scales (6,5), (6,5,4), and (6,5,4,3), respectively, with the analyzing wavelets. The error in reconstruction is very small if the replacement takes place at scales greater than the characteristic scale; the error increases significantly as the replacement takes place at the finer scales. In the general case, interactions between the neighboring objects will distort the wavelet transform response. However, even for complex signals, each object will eventually yield the (approximate) impulse response at the appropriate scale determined by the size of the object.

In order to find the characteristic scale of an image object in the 1–D case, one can measure the correlation between the wavelet transform of the object and the basis function at any given scale. At each location $(x,y)$, one can then select the finest scale at which the correlation exceeds some threshold. We will now proceed to examine the detection of a scale-space cell in two dimensions.

## 2.3 Saliency Detection Algorithm in Two Dimensions

In the two-dimensional case, the characteristic scale $I(\Theta)$ may be different for any particular orientation $\Theta$ of a 1–D cross-section through an object. Any object other than a circular one (disc) will become a point at different scales in different directions, e.g., an elongated object, occluded object, etc. In this case, the object will extend over several SSC's at any scale $j < \max_\Theta\{I(\Theta)\}$. Therefore, in the 2–D case, we apply the 1–D procedure in a number of directions and search for clusters of 1–D centers, as shown in Figure 3(a). Our entire procedure for detecting the SSC's in an image therefore consists of the following four steps [18]:

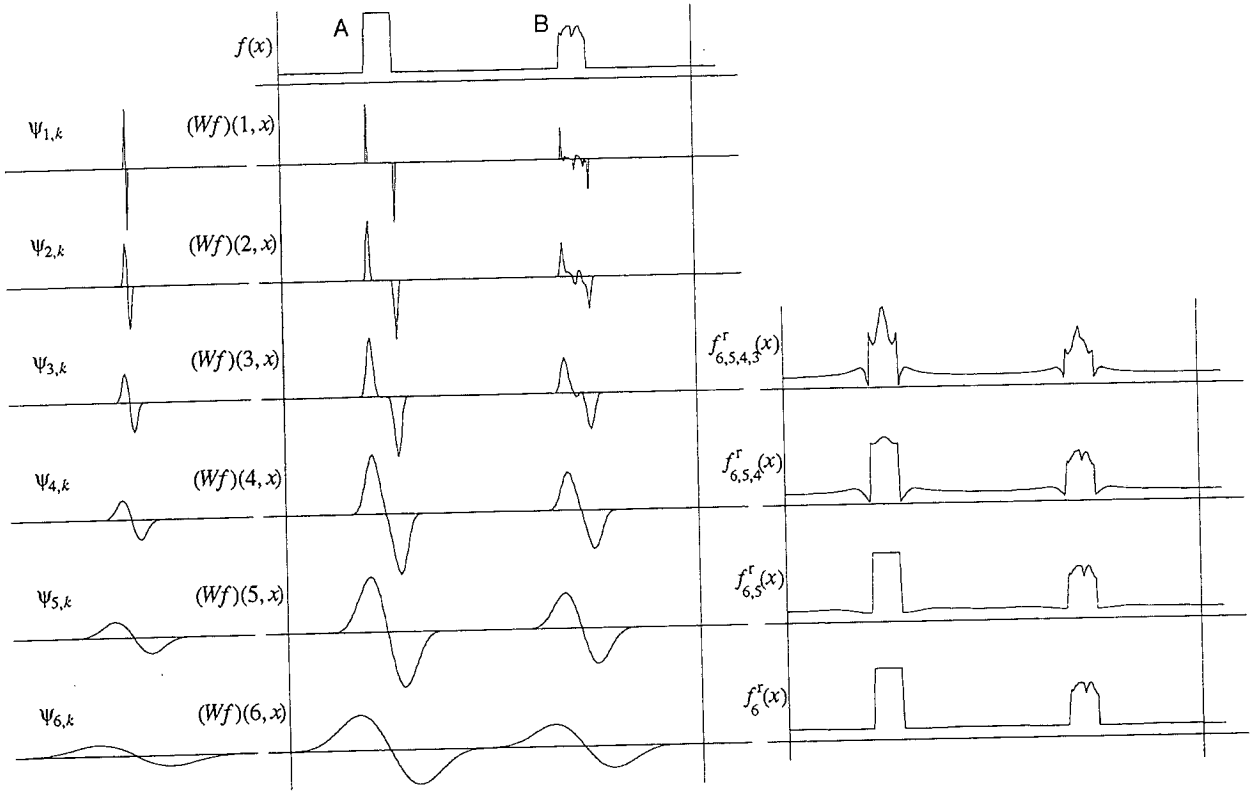
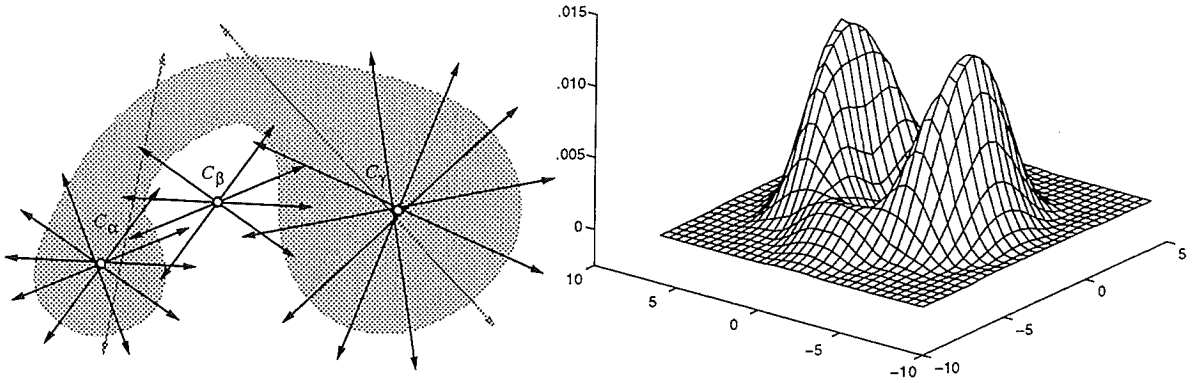



Figure 2: An illustration of the object's characteristic scale. The left column shows the analyzing wavelets $\psi_{i,k}$ at the corresponding scales. The middle column shows two examples (A and B) of a function $f(x)$ (top) and their wavelet transforms (below). Beginning with the characteristic scale $I = 4$, the $(W\cdot)(I + j, x)$ takes the shape of $\psi_{I+j,k}, j = 1, 2, ....$ The right column shows the reconstruction of the original function by replacing the $(W\cdot)(i, x)$ with the $\psi_{i,k}$.

**Step 1—Wavelet Transform:** Compute the wavelet pyramid of an image with $\ell$ dyadic scales using oriented quadrature bandpass filters tuned to 16 different orientations, i.e. $\Theta = 0°, 22.5°, 45°, ..., 337.5°$. See [26] for a detailed derivation and description of computing the wavelet pyramid using steerable basis filters.

**Step 2—Local Energies:** Compute the oriented local energies using the equation:

$$E(\Theta, s, x, y) = \left[G^{\Theta}(s, x, y)\right]^2 + \left[H^{\Theta}(s, x, y)\right]^2 \tag{2}$$




(a) (b)

Figure 3: (a) The clusters formed by the centers of the 1–D SSCs associated with the cross-sections through an object. (b) One of the filter kernels $\vartheta(\Theta, x, y)$ used in computing the saliency of the SSC's (Eq. (3)). The kernel is obtained by computing the oriented energy at $\Theta = 0°$ for a disc.

where $G^\Theta(s, x, y)$ and $H^\Theta(s, x, y)$ are the outputs of a quadrature pair of analyzing wavelet filters at the scale-space coordinate $(s, x, y)$, oriented at the angle $\Theta$. For each image point, 16 different oriented local energies are computed.

**Step 3—Saliency Maps:** Compute $\ell$ saliency maps. The saliency of each particular SSC is computed using the convolution:

$$\text{saliency SSC}(s, x, y) = \sum_{\Theta} [E(\Theta, s, x, y) * \vartheta(\Theta, x, y)] \tag{3}$$

where $\vartheta(\Theta, x, y)$ is the filter kernel obtained by computing the sum of the squared impulse responses of the two analyzing wavelet filters $G^\Theta(s, x, y)$ and $H^\Theta(s, x, y)$, as shown in Figure 3(b). As discussed above, circular shape has the highest saliency as measured by this scheme.

**Step 4—Peaks in Saliency Maps:** Moving from finer to coarser scales at every location, we select the first saliency map for the which a peak (local maximum) at that location exceeds a given threshold. By using a series of oriented 1-D filters to detect

the characteristic scale, we can detect objects that are not perfectly circular in shape. For example, if a non-circular shape's variation in diameter does not reach neighboring scales above or below the current scale, then a circularly-symmetric filter, such as that used by Crowley [5, 4, 6], will give a weak response for the shape. In our approach, however, the 1-D filters are slightly adjusted in width (bounded by neighboring scales). The result is a cluster of oriented peaks from which we compute the 2-D shape's location as the centroid of these peaks. The salience of the 2-D shape is computed as the sum of the oriented saliencies of the oriented peaks near this centroid. Finally, we apply a non-maximum suppression process to eliminate closely overlapping salient SSC's at each scale.

The contents of each scale-space cell (SSC) is a 2-D matrix of wavelet coefficients. The size of this matrix is invariant to both the scale at which the SSC is detected and the complexity of the shape contained in the SSC's corresponding image region. In the current implementation, only the scale, position, and saliency of a SSC is exploited during the matching of two saliency maps. However, one could include the actual content of the SSC as specified by the wavelet coefficient matrix. The fact that all the SSC matrices are self-similar and small (in our case, $16 \times 16$) means that efficient comparisons can be made between SSC's at different scales.

Figure 4 illustrates how a complex object's saliency map (a) is largely invariant to scaling (b), translation (c), image plane rotation (d), and limited rotation in depth (e), where the illuminated left side of the face exhibits little change in its saliency map. Circles in the image correspond to scale-space cells, while their intensity is proportional to the their saliency. Note that the size of the circle appears to be slightly larger than its corresponding image feature. This is due to the fact that the size of the circle is determined by the largest extent of the filter shown in Figure 3(b), i.e., where the response approaches zero. For different images taken under different conditions, there cannot be true invariance in the sense that salient regions are identical in different views. However, approximate invariance suffices for the recognition scheme that we propose below.

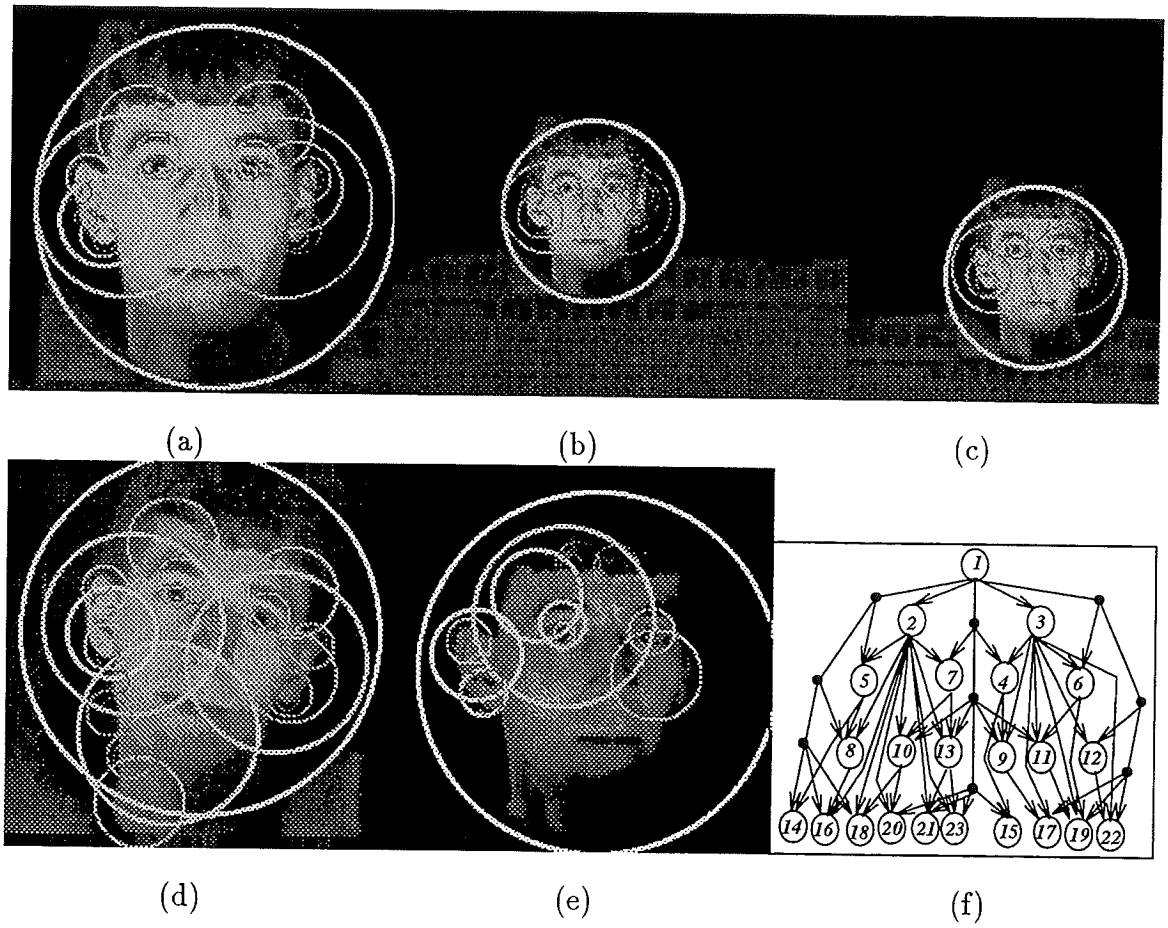



Figure 4: Extracting the most salient SSC's in an image: (a) original image and its saliency map; (b) scale invariance; (c) translation invariance; (d) image rotation invariance; (e) invariance to rotation in depth (illuminated left side of face exhibits little change in its saliency map); and (f) the saliency map graph (SMG) of the original image in (a).

## 2.4 Limitations of the Representation

Under normal circumstances, an object (or one of its component features) should produce a peak at its corresponding location in the saliency map. However, there are several exceptions which do not properly fit within the SSC framework. The evolution of the saliency map at a single scale as the object becomes degenerate, in the sense of the SSC framework, is shown in Figure 5. The exceptions arise due to wavelet transform scale sub-sampling, crowded objects, and elongated objects. As expected, combinations of these will create even more difficulties.

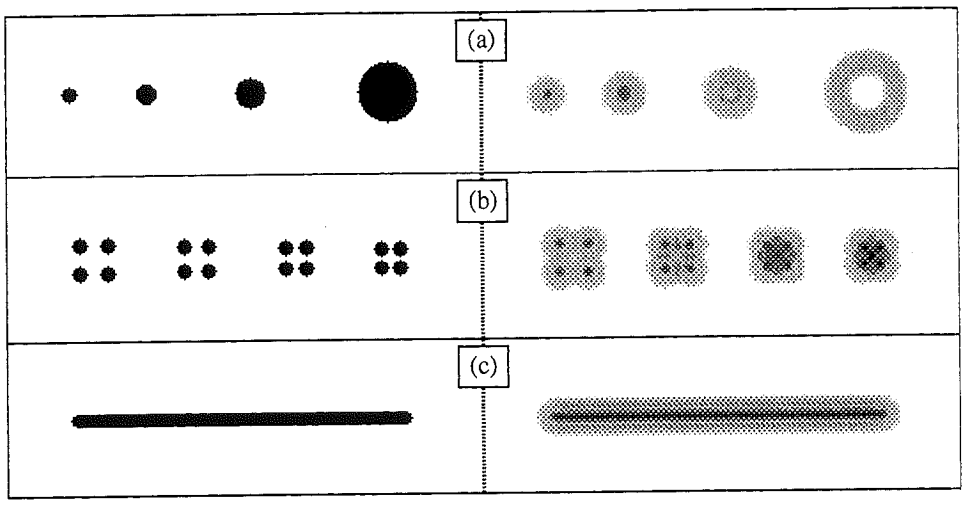



Figure 5: The evolution of the saliency map at a single scale for the exception cases. The left side shows the original images and the right side shows the corresponding saliency maps at one scale. (a) As the object size increases, a peak in the saliency map corresponding to the object turns into a "crater." (b) As the objects approach closer to one another, the "anti-object" between them becomes the most salient. (c) Instead of peak(s) in the saliency map, elongated objects produce "mountain ridges."

As shown in Figure 5(a), the saliency peak detector will find one salient region for the object at the left side, whereas it will find several salient regions for larger objects on the right side. This effect will occur when the size of an object is intermediate between two (octave) scales, and can be minimized by increasing the number of scales. In Figure 5(b), the detector will find four salient regions for the distant objects at left, but will find only one region in the center of the four close objects at right. This phenomenon occurs when a set of objects enclose a compact background region; the detector cannot separate figure from ground. One could argue that such a regular pattern represents a form of texture that should be treated as a single object. Again, if there were more scales, this composite object would be detected at a coarser scale. Finally, in Figure 5(c), the detector will find several salient regions positioned along the elongated object. In this case, a salient region grouper could search for a string of co-curvilinear SSC's at a given scale and group them into a composite structure which could easily be accommodated by our graph representation and matching

Given this formulation of the mapping, $f$, we define the error of $f$ to be:

$$\mathcal{E}(f) = \varepsilon \sum_{u \in V_1} \sum_{v \in V_2} M_{u,v}\, \omega(u,v)\, |s(u) - s(v)| + (1-\varepsilon)\left( \sum_{u \in V_1} (1 - \sum_{j \in V_2} M_{u,j}) s(u) + \sum_{v \in V_2} (1 - \sum_{i \in V_1} M_{i,v}) s(v) \right) \quad (4)$$

where $\varepsilon = |\mathbf{1}^t M(f) \mathbf{1}|/|V_1|$ represents the ratio of the number of matched vertices to the number of vertices ($|V_1|$) in the model SMG (with $\mathbf{1}$ the identity vector of the appropriate dimension) and $s(v)$ denotes the strength of region $v$ in its saliency map. For the SMG topological similarity algorithm, defined in Section 3.3, $\omega(u,v)$ is always one, while for the SMG geometrical similarity algorithm, defined in Section 3.4, $\omega(u,v)$ represents the Euclidean distance between the centers of the regions, $u$ and $v$. Clearly, in the case of perfect similarity, $\mathcal{E}(f) = 0$, while $\mathcal{E}(f)$ will be $\sum_{u \in V_1} s(u) + \sum_{v \in V_2} s(v)$ if there is no match ($\mathbf{1}^t |M(f)| \mathbf{1} = 0$).

## 3.3 A Matching Algorithm Based on Topological Similarity

In this section, we describe an algorithm which finds an approximate solution to the SMG similarity problem. The focus of the algorithm is to find a minimum weight matching between vertices of $G_1$ and $G_2$ which lie in the same level. Our algorithm starts with the vertices at level 1. Let $A_1$ and $B_1$ be the set of vertices at level 1 in $G_1$ and $G_2$, respectively. We construct a complete weighted bipartite graph $G(A_1, B_1, E)$ with a weight function defined for edge $(u,v)$ ($u \in A_1$ and $v \in B_1$) as $w(u,v) = |s(v) - s(u)|$.[2] Next, we find a maximum cardinality, minimum weight matching $M_1$ in $G$ using [9]. All the matched vertices are mapped to each other; that is, we define $f(x) = y$ if $(x,y)$ is a matching edge in $M_1$.

The remainder of the algorithm proceeds in phases as follows, as shown in Figure 6. In phase $i$, the algorithm considers the vertices of level $i$. Let $A_i$ and $B_i$ be the set of vertices of level $i$ in $G_1$ and $G_2$, respectively. Construct a weighted bipartite graph $G(A_i, B_i, E)$ as follows: $(v,u)$ is an edge of $G$ if either of the following is true: (1) Both $u$ and $v$ do not have any parent in $G_1$ and $G_2$, respectively, or (2) They have at least one matched parent

[2] $G(A, B, E)$ is a weighted bipartite graph with weight matrix $W = [w_{ij}]$ of size $|A| \times |B|$ if, for all edges of the form $(i,j) \in E$, $i \in A$, $j \in B$, and $(i,j)$ has an associated weight $= w_{i,j}$.

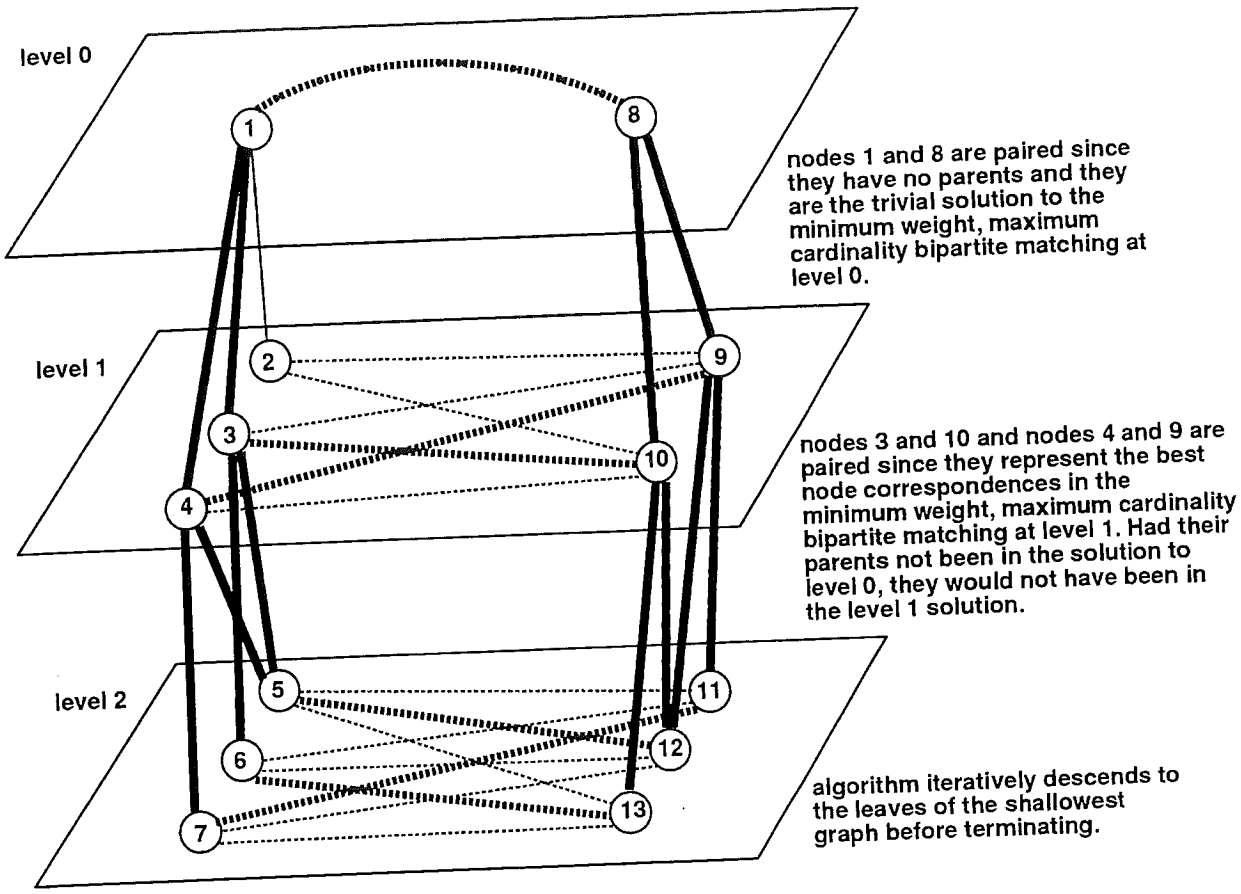



Figure 6: Illustration of the SMGBM Algorithm (see text for explanation).

of depth less than $i$; that is, there is a parent $p_u$ of $u$ and $p_v$ of $v$ such that $(p_u, p_v) \in M_j$ for some $j < i$. We define the weight of the edge $(u, v)$ to be $|s(u) - s(v)|$. The algorithm finds a maximum cardinality, minimum weight matching in $G$ and proceeds to the next phase.

The above algorithm terminates after $\ell$ phases, where $\ell$ is the minimum number of scales in the saliency maps (or SMG's) of two graphs. The partial mapping $M$ of SMG's can be simply computed as the union of all $M_i$'s for $i = 1, \ldots, \ell$. Finally, using the error measure defined above, we compute the error of the partial mapping $M$. Each phase of the algorithm requires simple operations with the time to complete each phase being dominated by the time to compute a minimum weight matching in a bipartite graph. The time complexity for finding such a matching in a weighted bipartite graph with $n$ vertices is $O(n^2\sqrt{n \log\log n})$ time, using the scaling algorithm of Gabow, Gomans and Williamson [10]. The entire procedure,

as currently formulated, requires $O(\ell n^2 \sqrt{n \log \log n})$ steps.

## 3.4 A Matching Algorithm Based on Geometric Similarity

The SMGBM similarity measure captured the structural similarity between two SMG's in terms of branching factor and node saliency similarity; no geometric information encoded in the SMG was exploited. In this section, we describe a second similarity measure, called SMG Similarity using an Affine Transformation (SMGAT), that includes the geometric properties (e.g., relative position and orientation) of the saliency regions.

Given $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, we first assume, without loss of generality, that $|V_1| \leq |V_2|$. First, as shown in Figure 7, the algorithm will hypothesize a correspondence between three regions of $G_1$, say $(r_1, r_2, r_3)$, and three regions $(r'_1, r'_2, r'_3)$ of $G_2$. The mapping $\{(r_1 \rightarrow r'_1), (r_2 \rightarrow r'_2), (r_3 \rightarrow r'_3)\}$ will be considered as a basis for alignment if the following conditions are satisfied:

- $r_i$ and $r'_i$ have the same level in the SMG's, for all $i \in \{1, \ldots, \ell\}$.
- $(r_i, r_j) \in E_1$ if and only if $(r'_i, r'_j) \in E_2$, for all $i, j \in \{1, \ldots, \ell\}$, which implies that selected regions should have the same adjacency structure in their respective SMG's.

Once regions $(r_1, r_2, r_3)$ and $(r'_1, r'_2, r'_3)$ have been selected, we solve for the affine transformation $(A, b)$, that aligns the corresponding region triples by solving the following system of linear inequalities:

$$\begin{bmatrix} x_{r_1} & y_{r_1} & 1 & 0 & 0 & 0 \\ x_{r_2} & y_{r_2} & 1 & 0 & 0 & 0 \\ x_{r_3} & y_{r_3} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{r_1} & y_{r_1} & 1 \\ 0 & 0 & 0 & x_{r_2} & y_{r_2} & 1 \\ 0 & 0 & 0 & x_{r_3} & y_{r_3} & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ b_1 \\ a_{21} \\ a_{22} \\ b_2 \end{bmatrix} = \begin{bmatrix} x_{r'_1} \\ x_{r'_2} \\ x_{r'_3} \\ y_{r'_1} \\ y_{r'_2} \\ y_{r'_3} \end{bmatrix}. \tag{5}$$

The affine transformation $(A, b)$ will be applied to all regions in $G_1$ to form a new graph $G'$. Next, a procedure similar to the minimum weight matching, used in the SMGBM is applied to

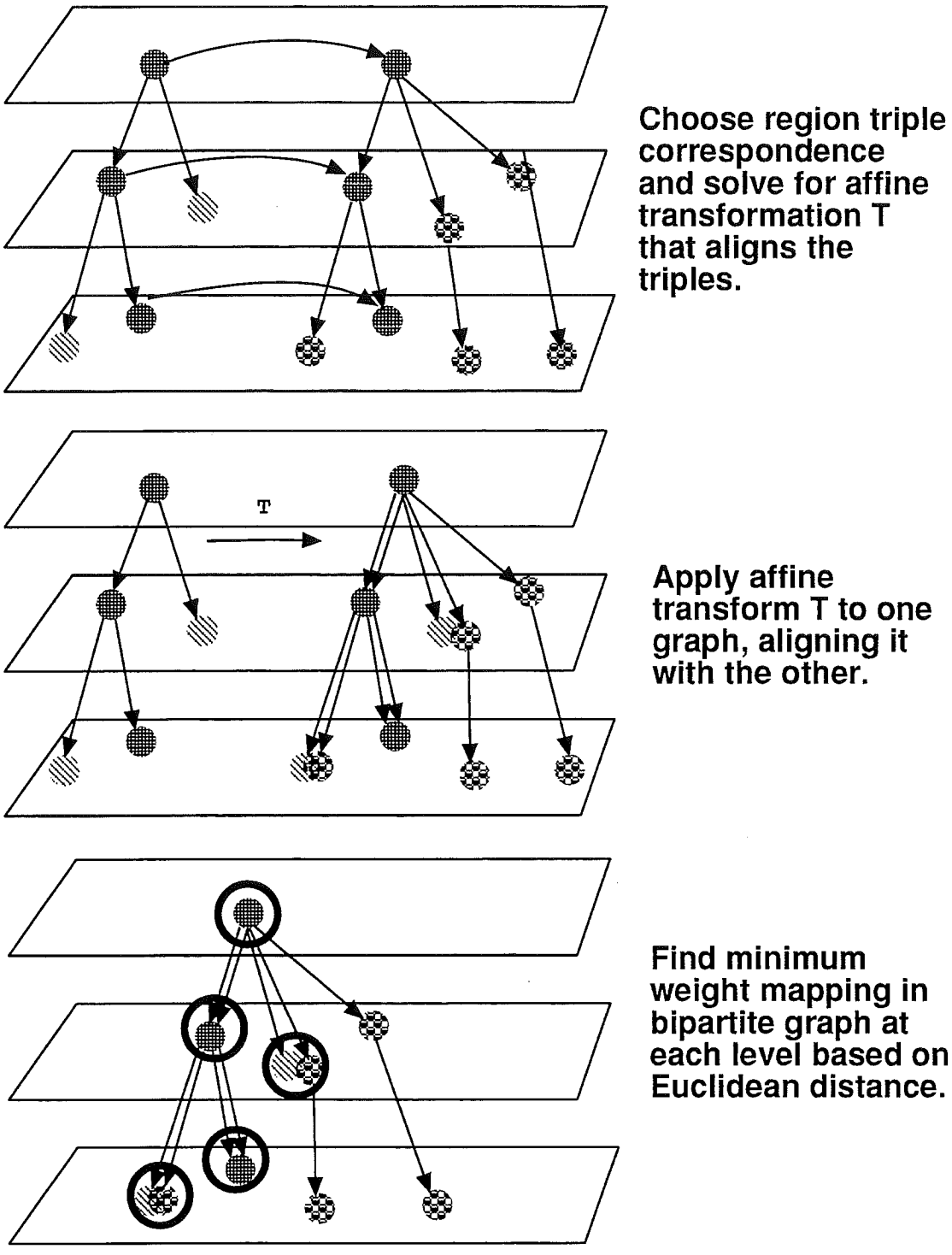



Figure 7: Illustration of the SMGAT Algorithm (see text for explanation)

the regions in graphs $G'$ and $G_2$. Instead of matching regions which have maximum similarity in terms of saliency, we match regions which have minimum Euclidean distance from each other. Given two regions $u$ and $v$, the distance between them can be defined as the $L_2$ norm of the distance between their centers, denoted by $d(u,v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}$. In a series of steps, SMGAT constructs weighted bipartite graphs $\mathcal{G}_i = (R_i, R_i', E_i)$ for each level $i$ of the two SMG's, where $R_i$ and $R_i'$ represent the set of vertices of $G'$ and $G_2$ at the $i$-th level, respectively. The constraints for having an edge in $E_i$ is the same as SMGBM: $(u,v)$ is an edge in $\mathcal{G}_i$ if either of the followings holds:

- Both $u$ and $v$ do not have any parents in $G'$ and $G_2$, respectively.
- They have at least one matched parent of depth less than $i$.

The corresponding edge will have weight equal to $w(u,v) = d(u,v)$. A maximum cardinality, minimum weight bipartite matching $M_i$ will be found for each level $\mathcal{G}_i$, and the partial mapping $f_{(A,b)}$ for the affine transformation $(A,b)$ will be formed as the union of all $M_i$'s. Finally, the error of this partial mapping $\mathcal{E}(f_{(A,b)})$ will be computed as the sum over each $E_i$ of the Euclidean distance separating $E_i$'s nodes weighted by the nodes' difference in saliency. Once the total error is computed, the algorithm proceeds to the next valid pair of region triples. Among all valid affine transformations, SMGAT chooses that one which minimizes the error of the partial mapping.

In terms of algorithmic complexity, solving for the affine transformation (eq. 5) takes only constant time, while applying the affine transformation to $G_1$ to form $G'$ is $O(\max(|V_1|, |E_1|))$. The execution time for each hypothesized pair of region triples is dominated by the complexity of establishing the bipartite matching between $G_2$ and $G'$, which is $O(\ell n^2 \sqrt{n \log\log n})$, for SMG's with $n$ vertices and $\ell$ scales. Although in the worst case, i.e., when both saliency map graphs have only one level, there are $O(n^6)$ pairs of triples. However, in practice, the vertices of an SMG are more uniformly distributed among the levels of the graph, greatly reducing the number of possible correspondences of base triples.

Although not yet implemented, we can reduce the complexity of the bipartite matching step by exploiting the fact that the edge weights of the bipartite graph represent the Eu-

clidean distance between the regions of two SMG's and satisfy the triangle inequality. The following algorithm can then be used to find the bipartite matching at the $i$-th level of $G'$:

1. Construct the Voronoi diagram of the vertices at the $i$-th level of $G'$ ($O(|R_i| \log |R_i|)$ time).
2. Project the regions in $R'_i$ into the plane of this Voronoi diagram.
3. In each Voronoi polytope, choose the closest region of $R'_i$ to the vertex defining the Voronoi polytope, if one exists ($O(\log |R_i|)$ time for each Voronoi region).
4. Update the Voronoi diagram by removing the matched vertices and their corresponding regions in $R'_i$ ($\Theta(\log |R_i|)$ time).
5. Repeat this process until either all the vertices at the $i$-th level of $G'$ are matched or the remaining vertices cannot be matched (due to path constraints).

The time complexity of the above procedure for level $i$ is $O(n^2 \log n)$ [22]. For $\ell$ levels, the total complexity is $O(\ell n^2 \log n)$, which compares favorably to the $O(\ell n^2 \sqrt{n \log \log n})$ complexity of our current SMGAT algorithm.

## 3.5 Limitations of the Matching Algorithms

There are two major limitations of both matching algorithms. First, since both algorithms seek a minimum weight, maximum cardinality matching in a bipartite graph that spans corresponding levels of two saliency map graphs, corresponding nodes in the two graphs must therefore lie at the same levels in their respective SMG's. This implies that a scene SMG cannot be vertically expanded or compressed relative to a model SMG. Furthermore, an image object that is detected at a scale different from that of its corresponding model object cannot be correctly matched.

To overcome this problem, consider a model SMG corresponding to a particular view of some object and let the initial scene SMG be exactly equal to the model SMG. Next, consider

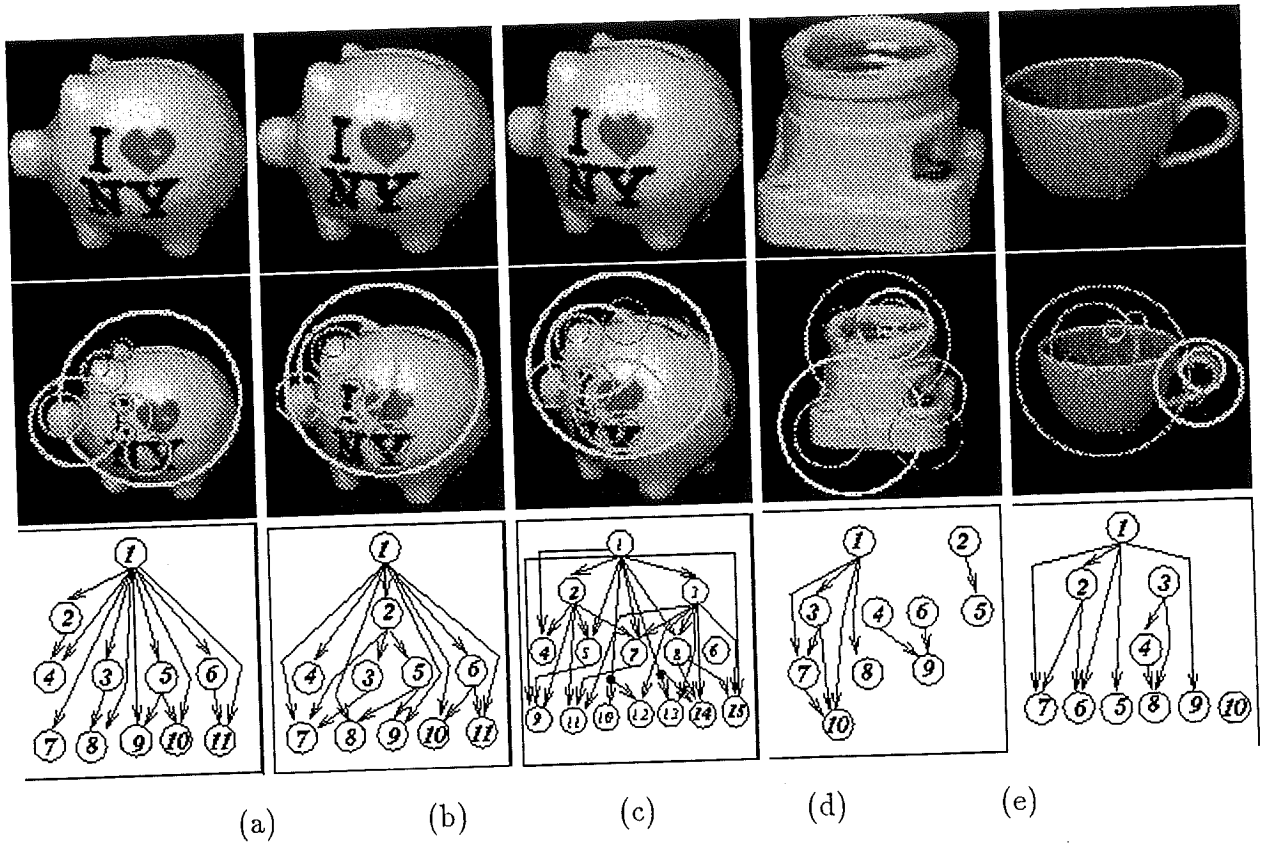
a perturbation of the scene SMG in which any scene SMG node can migrate up or down a small number of levels, $k$, provided that the scene SMG topology remains intact, i.e., same parent-child relationships with parents and/or children changing levels. For a fixed, small $k$, the bandwidth of the bipartite graph mapping solution will increase from 1 to $2k+1$. In other words, the bipartite graph previously generated at each level will now encompass nodes at neighboring levels. The resulting complexity of both algorithms will be the same except for a constant scaling factor to account for the increased (constant) number of nodes in each bipartite graph.

The restriction that corresponding nodes lie at the same level or scale has an important implication for matching cluttered scenes. If the scale of background objects is comparable to the object being recognized, the saliency map graph corresponding to the scene is approximately the saliency map corresponding to the object with additional nodes added to one or more levels. In this case, our assumption that corresponding nodes exist at the same level is not violated. However, if a background object dominates the object being recognized, the effect will be to "push" the object down to a finer scale while the background object occupies the coarser scales. This migration of the target object would violate our assumption that corresponding SMG nodes lie at the same scale.

If we assume that some model SMG, consisting of $\ell$ levels, occupies any $\ell$ continuous levels of a scene SMG, then to overcome this second problem requires that we apply either algorithm to each level of the scene SMG. This will mean that the complexity of both algorithms will increase by a multiplicative factor of $h$, the number of scales in the scene SMG. In the experiments reported in the following section, we assume a bipartite graph of bandwidth 1 and assume that any background objects do not dominate the target object.

# 4 Experiments

To illustrate our approach to shape representation and matching, we apply it to a database of model object views generated by Murase and Nayar at Columbia University. Views of each of the 20 objects are taken from a fixed elevation every 5 degrees (72 views per object) for



(a) (b) (c) (d) (e)

Figure 8: A sample of views from the database: top row represents original images, second row represents saliency maps, while third row represents saliency map graphs.

a total of 1440 model views. The top row of images in Figure 8 shows three adjacent model views for one of the objects (piggy bank) plus one model view for each of two other objects (bulb socket and cup). The second row shows the computed saliency maps for each of the five images, while the third row shows the corresponding saliency map graphs. The time to compute the saliency map averaged 156 seconds/image for the five images on a Sun Sparc 20, but can be reduced to real-time on a system with hardware support for convolution, e.g., a Datacube MV200. The average time to compute the distance between two SMG's is 50 ms using SMGBM, and 1.1 second using SMGAT (an average of 15 nodes per SMG).

| Algorithm | % Hit | % Miss right object | % Miss wrong object |
|---|---|---|---|
| SMGBM | 89.0 | 8.4 | 2.6 |
| SMGAT | 96.6 | 2.9 | 0.5 |

Table 3: An exhaustive test of the two matching algorithms. For each image in the database, the image is removed from the database and compared, using both algorithms, to every remaining image in the database. The closest matching image can be either one of its true neighboring views, a different view belonging to the correct object, or a view belonging to a different object.

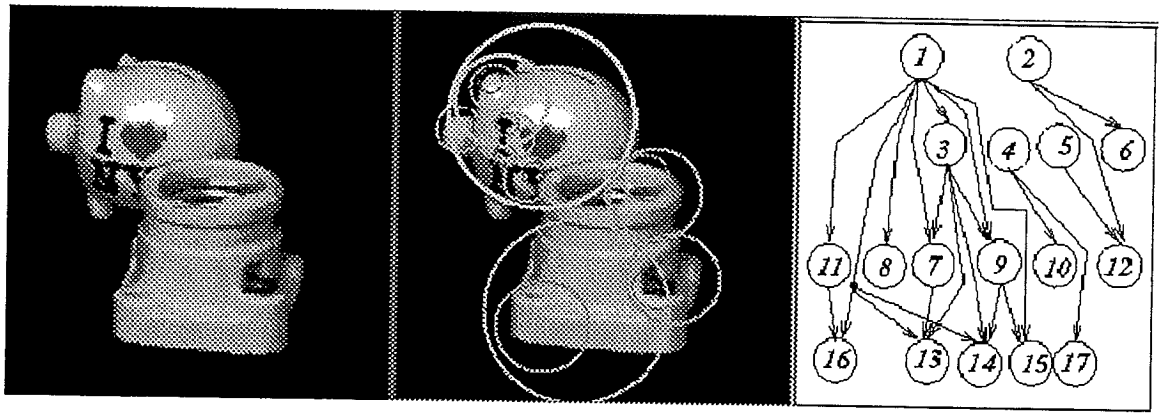



Figure 10: Occluded Object Matching: (a) original image; (b) saliency map; and (c) saliency map graph

in Table 5. In this case, the closest view is the correct view (Figure 8(d)) of the socket.

In a second occlusion experiment, consider the duck occluding the toy cat, as shown in Figure 11. The closest matching view in the database is the correct view of the duck (Figure 9(a)). After removing the scene SMG subgraph corresponding to the duck, the remaining subgraph was matched to the entire database, as shown in Table 7. The closest image is the correct view (Figure 9(d)) of the cat.

| Algorithm | 8(a) | 8(b) | 8(c) | 8(d) | 8(e) |
|---|---|---|---|---|---|
| SMGBM | 9.56 | 3.47 | 8.39 | 12.26 | 14.72 |
| SMGAT | 24.77 | 9.29 | 21.19 | 30.17 | 33.61 |

Table 4: Distance of Figure 10(a) to other images in Figure 8. The correct piggy bank view (Figure 8(b)) is the closest matching view.

| Algorithm | 8(a) | 8(b) | 8(c) | 8(d) | 8(e) |
|---|---|---|---|---|---|
| SMGBM | 12.42 | 14.71 | 14.24 | 4.53 | 9.83 |
| SMGAT | 18.91 | 20.85 | 17.08 | 7.19 | 15.44 |

Table 5: Distance of Figure 10(a) (after removing from its SMG the subgraph corresponding to the matched piggy back image) to other images in Figure 8.

| Algorithm | 9(a) | 9(b) | 9(c) | 9(d) | 9(e) |
|---|---|---|---|---|---|
| SMGBM | 22.71 | 29.64 | 33.97 | 30.57 | 62.11 |
| SMGAT | 39.16 | 47.92 | 66.04 | 85.19 | 105.72 |

Table 6: Distance of Figure 11 to other images in Figure 9.

| Algorithm | 9(a) | 9(b) | 9(c) | 9(d) | 9(e) |
|---|---|---|---|---|---|
| SMGBM | 78.68 | 62.41 | 71.27 | 27.59 | 51.03 |
| SMGAT | 75.92 | 81.39 | 68.41 | 44.37 | 90.29 |

Table 7: Distance of Figure 11 (after removing from its corresponding SMG the subgraph corresponding to the matched duck image) to the other images in Figure 9.

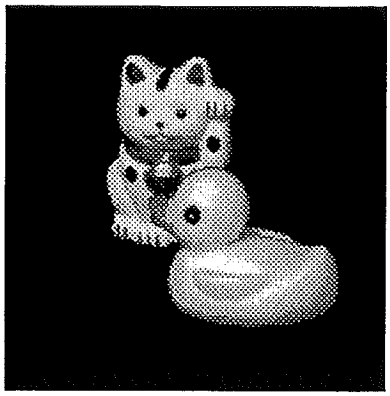


Figure 11: Image of occluded object used in second occlusion experiment

## 4.3 An Analysis of Viewpoint Invariance

In a view-based 3-D object recognition system, an object is represented by a collection of views. The more viewpoint-invariant an image representation is, the fewer the number of views needed to represent the object. In the above experiments, we computed the saliency map graphs for the full set of 72 views for each of the 20 objects. In this section, we explore the viewpoint invariance of our representation by considering a smaller sample of views for one of our objects.

Our experiment, as shown in Figure 12, consists of successively removing every second view (model SMG's) of a given object (in this case, the piggy bank) and computing the distance, using both SMGBM and SMGAT, between each removed view to the remaining views. Thus, at the first iteration, we will remove every second view from the original set of 72 views, leaving 36 views of the model object. Each of the 36 views that was removed will then be compared to each of the 36 remaining model views. If the closest matching model view is adjacent to the removed view's position in the original set of 72 views, then one can argue that the intermediate view (that was removed) is extraneous. At the next iteration, we remove every second view from the 36 model views and repeat the experiment with the 18 removed views.[3]

The results are shown in Table 8. For example, when leaving out 36 views, 91% of the

[3]The $n$ views removed at step $\ell$ are maximally distant from the $n$ remaining views; there is no need to match the views removed at step $\ell - 1$ to the views remaining at step $\ell$.

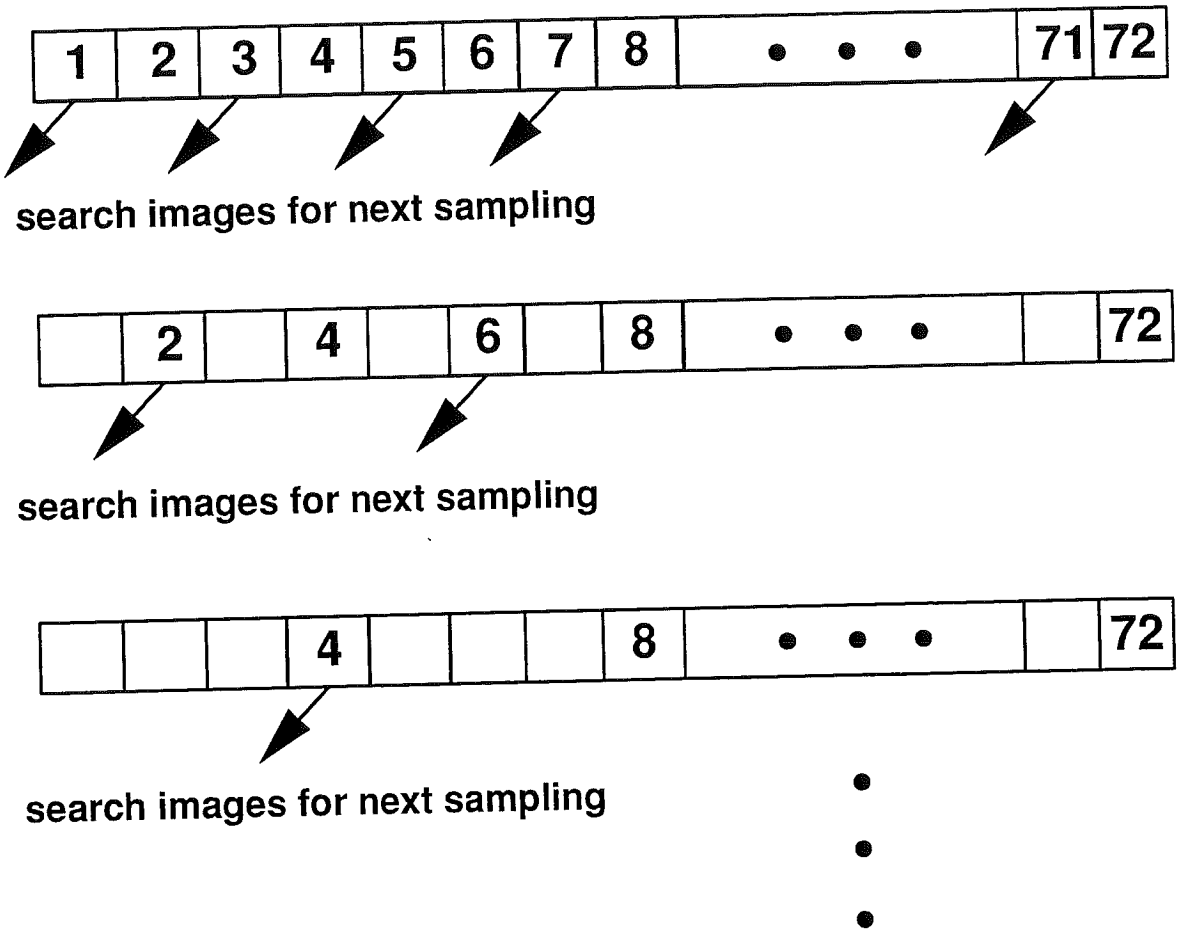



Figure 12: An Experiment Exploring the Viewpoint Invariance of the SMG Representation and Matching Algorithms

| Views in Tree | 36 | 18 | 9 |
|---|---|---|---|
| SMGBM % | 91 | 50 | 35 |
| SMGAT % | 99 | 84 | 61 |

Table 8: Evaluating Viewpoint Invariance of the SMG Representation. The first row indicates the number of model views remaining in the model view set for the piggy bank object after removing every second view. The second and third rows indicate the percentage of SMGBM-based and SMGAT-based searches, respectively, between each of the removed views and the remaining model views that result in a "closest" view that is adjacent to the removed view.

SMGBM searches (using a removed view) resulted in a closest view that is adjacent to the removed view at the next level up (72 views), while for SMGAT, 99% of the searches were successful. Furthermore, this percentage gradually declines for SMGAT and rapidly declines for SMGBM. As one might expect, when geometric information is included in the search, neighboring views of a test view exhibit the least geometric distortion. For the SMGBM algorithm, however, the topological structure of a test view may, in fact, be similar to other views of the object despite geometric differences.

With the proper indexing structure, it is clear that in a recognition framework, the number of candidates returned from a topological index will be higher than that returned from a geometric index, given the ambiguity inherent in a topological index. On the other hand, shape deformations within an object's class may be accommodated by SMGBM and not by SMGAT. Finally, it must be pointed out that the above analysis was performed on only one object. Although we would expect the same trend to occur with other model object view sets, the percentages will vary with the shape and appearance of the object. For example, for an object with many degenerate views, we would expect the percentages to fall when a sample lies directly on a degenerate view. We are currently conducting more comprehensive experiments in order to to predict what kind of view sampling resolutions are appropriate for each algorithm.

## 4.4 Limitations of the Experiments

The approach presented in this paper has not addressed the indexing problem. For the experiments, each "query" view was compared to each and every model view to return the closest matching view. In current work, we are exploring the use of recovered local SMG structure (SMG subgraphs covering local regions in the image) to index into the database of model views and return objects whose model view trees have similar structure at their leaves. In addition, we are exploring hierarchical representations of the model views corresponding to a given object, leading to a more efficient ($O(\log n)$) search of an object's model views than the current linear search. The evaluation of our approach is also limited in that by using the Columbia University image database, we were unable to change the lighting conditions, scale, etc., of the images. In future work, we plan to construct our own image database, allowing us to more effectively evaluate the transformation invariance of our representation.

# 5 Conclusions

There is a gap in the view-based object recognition literature between the image-based systems and the feature-based systems. While the image-based systems have been shown to work with complex objects, e.g., faces, they are highly sensitive to occlusion, scale, and deformation. The feature-based systems, on the other hand, rely on highly sensitive feature extraction processes. We have introduced an image representation that fills this gap. Our saliency map graph offers a robust, transformation invariant, multiscale representation of an image that not only captures the salient image structure, but provides the locality of representation required to support occluded object recognition. We have presented two graph matching algorithms, SMGBM and SMGAT, that offer an effective mechanism for comparing the topological and geometric structure, respectively, of a test image SMG and a database image SMG.

Our graph matching formulation, in terms of topological and geometric similarity, is applicable to any multiscale image representation, e.g., a Laplacian pyramid, which can be mapped to a vertex-weighted, directed acyclic graph. We are not only seeking to improve our

saliency map construction, but are exploring other multiscale image representations within this framework. We are also embedding our matching algorithms in an object recognition system that uses SMG subgraphs as an indexing structure.

## Acknowledgements

We gratefully acknowledge Columbia University's Shree Nayar for providing us with the database of model views. We also gratefully acknowledge the support of the National Science Foundation.

## References

[1] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–277, San Francisco, CA, June 1996.

[2] J. Burns and L. Kitchen. Recognition in 2D images of 3D objects from large model bases using prediction hierarchies. In *Proceedings, International Joint Conference on Artificial Intelligence*, pages 763–766, Milan, Italy, 1987.

[3] P. J. Burt. Attention Mechanisms for Vision in a Dynamic World. In *Proceedings of the International Conference on Pattern Recognition, Vol.1*, pages 977–987, The Hague, The Netherlands, 1988.

[4] J. Crowley and A. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):156–169, March 1984.

[5] J. L. Crowley. A Multiresolution Representation for Shape. In A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, pages 169–189. Springer Verlag, Berlin, 1984.

[6] J. L. Crowley and A. C. Sanderson. Multiple Resolution Representation and Probabilistic Matching of 2–D Gray–Scale Shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):113–121, January 1987.

[7] S. Dickinson, A. Pentland, and A. Rosenfeld. 3-D shape recovery using distributed aspect matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):174–198, 1992.

[8] G. G. E. Mjolsness and P. Anandan. Optimization in model matching and perceptual organization. *Neural Computation*, 1:218–229, 1989.

[9] E. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[10] H. Gabow, M. Goemans, and D. Williamson. An efficient approximate algorithm for survivable network design problems. *Proc. of the Third MPS Conference on Integer Programming and Combinatorial Optimization*, pages 57–74, 1993.

[11] K. Ikeuchi and T. Kanade. Automatic generation of object recognition programs. *Proceedings of the IEEE*, 76:1016–1035, 1988.

[12] M. Jägersand. Saliency Maps and Attention Selection in Scale and Spatial Coordinates: An Information Theoretic Approach. In *Proceedings of the 5th International Conference on Computer Vision*, pages 195–202, Boston, MA, June 1995.

[13] J. Kobler. *The Graph Isomorphism Problem: Its Structural Complexity.* Birkhauser, Boston, 1993.

[14] J. Koenderink and A. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[15] A. Leonardis and H. Bischoff. Dealing with occlusions in the eigenspace approach. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 453–458, San Francisco, CA, June 1996.

[16] T. Lindeberg. Detecting Salient Blob–Like Image Structures and Their Scales With a Scale–Space Primal Sketch—A Method for Focus–of–Attention. *International Journal of Computer Vision*, 11(3):283–318, December 1993.

[17] S. Mallat and W. L. Hwang. Singularity Detection and Processing with Wavelets. *IEEE Transactions on Information Theory*, 38(2):617–643, March 1992.

[18] I. Marsic. Data–Driven Shifts of Attention in Wavelet Scale Space. Technical Report CAIP–TR–166, CAIP Center, Rutgers University, Piscataway, NJ, September 1993.

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

[20] B. Olshausen, C. Anderson, and D. V. Essen. A Neurobiological Model of Visual Attention and Invariant Pattern Recognition Based on Dynamic Routing of Information. *Journal of Neurosciences*, 13(11):4700–4719, November 1992.

[21] A. Pope and D. Lowe. Learning object recognition models from images. In *Proceedings, IEEE International Conference on Computer Vision*, pages 296–301, Berlin, May 1993.

[22] F. Preparata and M. Shamos. *Computational Geometry.* Springer-Verlag, New York, NY, 1985.

[23] R. P. N. Rao, G. J. Zelinsky, M. M. Hayhoe, and D. H. Ballard. Modeling Saccadic Targeting in Visual Search. In D. Touretzky, M. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 830–836. MIT Press, Cambridge, MA, 1996.

[24] C. Schmid and R. Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 872–877, San Francisco, CA, June 1996.

[25] E. Simoncelli, W. Freeman, E. Adelson, and D. Heeger. Shiftable multi-scale transforms. *IEEE Transactions on Information Theory*, 38(2):587–607, 1992.

[26] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable Multiscale Transforms. *IEEE Transactions on Information Theory*, 38(2):587–607, March 1992.

[27] B. Takaĉs and H. Wechsler. A Dynamic and Multiresolution Model of Visual Attention and Its Application to Facial Landmark Detection. *Computer Vision and Image Understanding*, (in press).

[28] J. K. Tsotsos. An Inhibitory Beam for Attentional Selection. In L. Harris and M. Jenkin, editors, *Spatial Vision in Humans and Robots*, pages 313–331. Cambridge University Press, Cambridge, UK, 1993.

[29] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[30] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992–1006, October 1991.