View-Based Object Matching

Ali Shokoufandeh
Dept. of Comp. Sci. and
Center for Cognitive Science
Rutgers University
New Brunswick, NJ 08903

Ivan Marsic
Dept. of Electrical
and Computer Eng.
Rutgers University
New Brunswick, NJ 08903

Sven J. Dickinson
Dept. of Comp. Sci. and
Center for Cognitive Science
Rutgers University
New Brunswick, NJ 08903

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.

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. 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. Most 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, e.g., [13]. In contrast to the feature-based approaches, whose success requires reliable segmentation, a number of image-based view-based recognition systems have emerged [8, 5, 10]. Although these image-based approaches have been shown to work on natural objects, they are sensitive to (one or more of) illumination changes, scaling, image rotation, depth rotation, or occlusion.

Coarse-to-fine image descriptions have much support in the computer vision community, e.g., [1, 6]. Most of these have been developed for attention purposes and, as a result, lose the detailed shape information required for object recognition. In a top-down recognition system, 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 [9]. Crowley presented an approach which is related to the approach presented in this paper, in which circular features are detected in a Laplacian pyramid [2]. The resulting features are linked to form a tree, with the similarity of two trees defined as the similarity of paths 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. 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. A more comprehensive version of this paper may be found in [11].

2 A Scale-Space Saliency Representation of an Image

The scale-space image representation that we have selected is based on a multiscale wavelet transform [12]. The advantage of the wavelet decomposition lies in its effective time (space)–frequency (scale) localization. In the output of the transform, 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, we select the *characteristic 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). Our procedure for detecting the SSC's in an image consists of the following four summarized steps [7], while Figure 1 illustrates the invariance of the SSC's; a more comprehensive explanation can be found in [11].

Step 1—Wavelet Transform: Compute the wavelet pyramid of an image with ℓ dyadic scales using oriented quadrature bandpass filters tuned to 16 different orientations, i.e. $\Theta = 0°, 22.5°, 45°, ..., 337.5°$. See [12] 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:

$$E(\Theta, s, x, y) = \left[G^{\Theta}(s, x, y)\right]^2 + \left[H^{\Theta}(s, x, y)\right]^2 \quad (1)$$

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 Θ. For each image point, 16 different oriented local energies are computed.

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

$$\text{saliency SSC}(s, x, y) = \sum_{\Theta} \left[E(\Theta, s, x, y) * \vartheta(\Theta, x, y)\right] \quad (2)$$

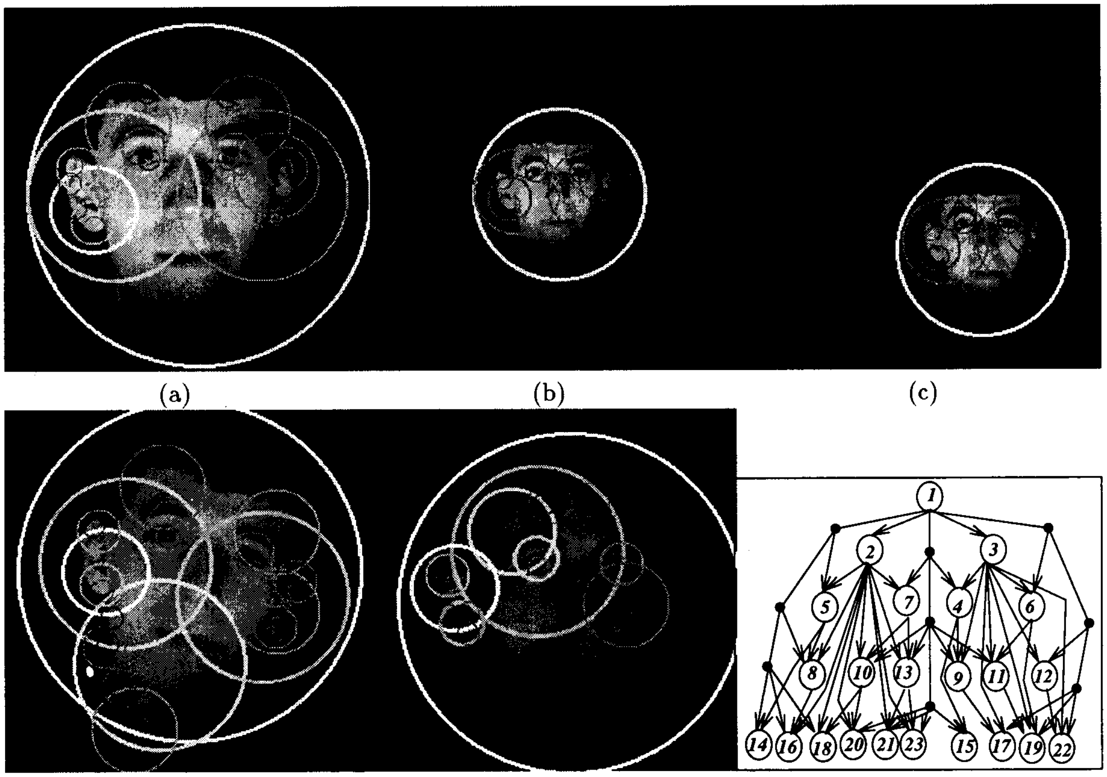
where $\vartheta(\Theta, x, y)$ is the filter kernel obtained by computing the sum of the squared impulse responses of the two analyzing wavelet filters $G^{\Theta}(s, x, y)$ and $H^{\Theta}(s, x, y)$.

Step 4—Peaks in Saliency Maps: Moving from finer to coarser scales at every location, we select the first saliency map for 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 [2], 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 computed saliency map can be represented as a hierarchical graph with nodes representing saliency regions and specifying region location (in the image), region size, region saliency, and scale level. More formally, we define the Saliency Map Graph (SMG) to be a directed acyclic graph $G = (V, E)$, with each saliency region r_i having a vertex v_i in V. (v_i, v_j) is a directed edge in E if and only if the scale level of region r_i is less than the scale level of region r_j, and the center of the region r_j lies in the interior of the region r_i. All the edges of G will therefore be directed from vertices at a coarser scale to vertices at a finer scale, as shown in Figure 1 (lower-right). Finally, to construct the database of object views, a set of views is obtained for each object from a fixed number of viewpoints (e.g., a regularly sampled tessellation of a viewing sphere centered at the object). For each view, the Saliency Map Graph is computed and stored in the database.

3 Matching Two Saliency Map Graphs

Given the SMG computed for an input image to be recognized and an SMG computed for a given model object image (view), we propose two methods for computing their similarity. In the first method, we compare only the topological or structural similarity of the graphs, a weaker distance measure designed to support limited object deformation invariance. In the second method, we take advantage of the geometrical information encoded in an SMG and strengthen the similarity measure to ensure geometric consistency, a stronger distance measure designed to support subclass or instance matching. It is imperative that each method support a measure of subgraph similarity in order to support occluded object matching.



(a) (b) (c)

(d) (e) (f)

Figure 1: Extracting the most salient SSC's in an image: (a) original image and its saliency map; (b) scale i ance; (c) translation invariance; (d) image rotation invariance; (e) invariance to rotation in depth (illumi left side of face exhibits little change in its saliency map); and (f) the saliency map graph (SMG) of the or image in (a).

3.1 Problem Formulation

Two graphs $G = (V, E)$ and $G' = (V', E')$ are said to be isomorphic if there exists a bijective mapping $f : V \longrightarrow V'$ satisfying, for all $x, y \in V$ $(x, y) \in E \Leftrightarrow (f(x), f(y)) \in E'$. To compute the similarity of two SMG's, we consider a generalization of the graph isomorphism problem, which we will call the *SMG similarity problem*: Given two SMG's $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and a partial mapping from $f : V_1 \longrightarrow V_2$, let $\mathcal{E}$ be a real-valued error function defined on the set of all partial mappings. We say that a partial mapping f is feasible if $f(x) = y$ implies that there are parents p_x of x and p_y of y, such that $f(p_x) = p_y$. Our goal is therefore to find a feasible mapping f which minimizes $\mathcal{E}(f)$.

3.2 Choosing a Suitable Error Function

The requirement of feasibility assures that the partial mapping preserves the path structure between G_1 and G_2. The error function incorporates two components: 1) the similarity of mapped nodes in terms of their topology, geometry, and salience; and 2) the degree to which model nodes are excluded from the mapping. Given two SMG's, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, and a partial mapping, $f : V_1 \longrightarrow V_2$, we define the mapping matrix $M(f)$ between G_1 and G_2, to be a $|V_1| \times |V_2|$, $\{0, 1\}$-matrix with $M_{u,v}$ equal to 1 if $u \in V_1$, $v \in V_2$ and $u = f(v)$, and 0 otherwise. Since f is a bijective mapping, $\sum_{v \in V_2} M_{u,v} \leq 1$ for all $u \in V_1$, and $\sum_{u \in V_1} M_{u,v} \leq 1$ for all $v \in V_2$. Given this formulation of the mapping f, we define its error to be:

$$\begin{aligned} \mathcal{E}(f) \quad = \quad & \varepsilon \sum_{u \in V_1, v \in V_2} M_{u,v}\ \omega(u,v)\ |s(u) - s(v)| + \\ & (1-\varepsilon) \sum_{u \in V_1, f(u) = \emptyset} s(u) \qquad (3) \end{aligned}$$

where $\varepsilon = |\mathbf{1}^t M(f) \mathbf{1}| / (|V_1| + |V_2|)$ represents the fraction of matched vertices (**1** denotes the identity vector), $f(.) = \emptyset$ for unmatched vertices, and $s(.)$ represents region saliency. For the SMG topological similarity, Section 3.3, $\omega(.,.)$ is always one, while for the SMG geometrical similarity, Section 3.4, it denotes the Euclidean distance between the regions.[1] A more comprehensive discussion on the error function can be found in [11].

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; illustrative examples can be found in [11]. 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 [3]. 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. 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 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 ℓ phases, where ℓ 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, Goemans and Williamson [4]. 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; illustrative examples can be found in [11].

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, 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:

$$\begin{bmatrix} x_{r_1} & y_{r_1} & 1 & 0 & 0 & 0 \\ x_{r_2} & y_{r_2} & 1 & 0 & 0 & 0 \\ x_{r_3} & y_{r_3} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{r_1} & y_{r_1} & 1 \\ 0 & 0 & 0 & x_{r_2} & y_{r_2} & 1 \\ 0 & 0 & 0 & x_{r_3} & y_{r_3} & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ b_1 \\ a_{21} \\ a_{22} \\ b_2 \end{bmatrix} = \begin{bmatrix} x_{r'_1} \\ x_{r'_2} \\ x_{r'_3} \\ y_{r'_1} \\ y_{r'_2} \\ y_{r'_3} \end{bmatrix} \qquad (4)$$

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 the regions in graphs G' and G_2. Instead of matching regions which have maximum similarity in terms of saliency,

[1] For perfect similarity $\mathcal{E}(f) = 0$, while $\mathcal{E}(f)$ will be $\sum_{u \in V_1} s(u)$ if there is no match.

[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 $= w_{i,j}$.

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. 4) 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 ℓ scales. 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. For an expanded discussion on complexity, including a Voronoi-based technique for reducing the complexity of the bipartite matching, see [11].

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 total of 1440 model views. The top row of images in Figure 2 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 seconds using SMGAT (an average of 15 nodes per SMG).

Algorithm	2(a)	2(c)	2(d)	2(e)
SMGBM	9.57	10.06	14.58	23.25
SMGAT	8.91	12.27	46.30	43.83

Table 1: Distance of Figure 2(b) to other images in Figure 2

4.1 Unoccluded Scenes

To illustrate the matching of an unoccluded image to the database, we compare the middle piggy bank image (Figure 2(b)) to the remaining images in the database. Table 1 shows the distance of the test image to the other images in Figure 2; the two other piggy bank images (Figures 2 (a) and (c)) were the closest matching views in the entire database. Table 1 also illustrates the difference between the two matching algorithms. SMGBM is a weaker matching algorithm, searching for a topological match between two SMG's. SMGAT, on the other hand, is more restrictive, searching for a geometrical match between the two SMG's. For similar views, the two algorithms are comparable; however, as two views diverge in appearance, their similarity as computed by SMGAT diverges more rapidly than their SMGBM similarity. Additional results can be found in [11].

In the second experiment, we compare every image to every other image in the database, resulting in over 1 million trials. There are three possible outcomes: 1) the image removed from the database is closest to one of its neighboring views of the correct object; 2) the image removed from the database is closest to a view belonging to the correct object but not a neighboring view; and 3) the image removed from the database is closest to a view belonging to a different object. The results are shown in Table 2. As we would expect, the SMGAT algorithm, due to its stronger matching criterion, outperforms the SMGBM algorithm. If we

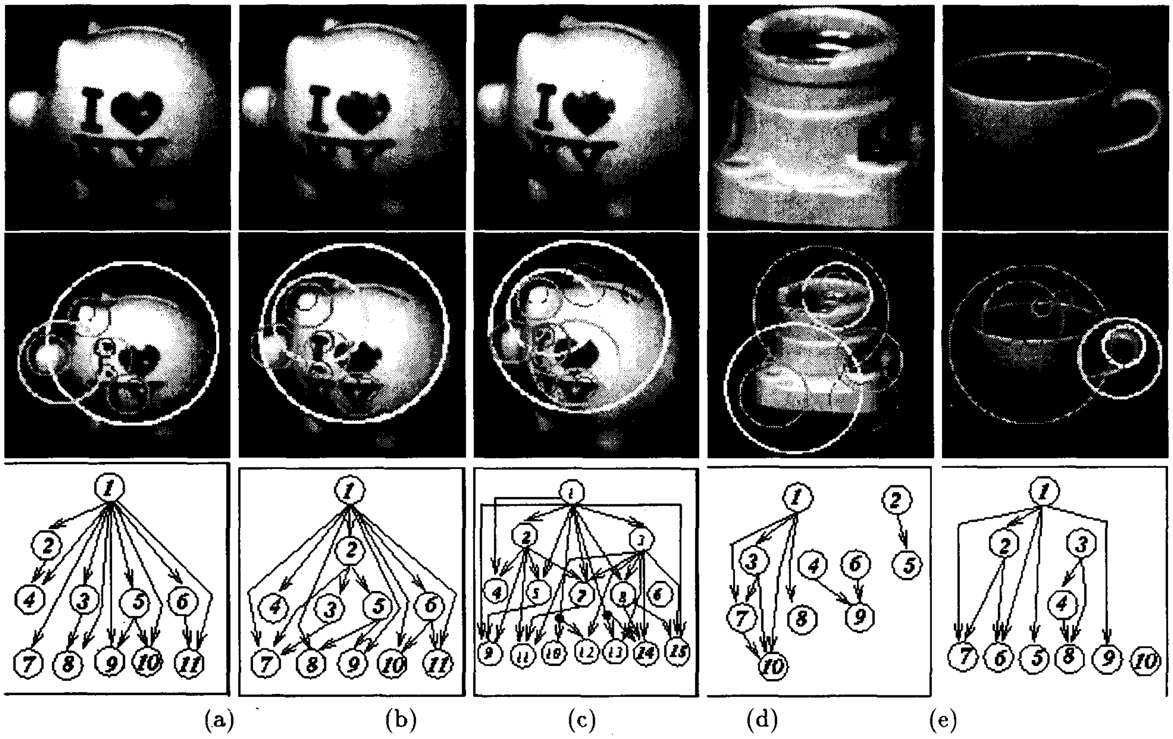


Figure 2: A sample of views from the database: top row represents original images, second row represents saliency maps, while third row represents saliency map graphs.

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

Table 2: Each image in the database 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.

include as a correct match any image belonging to the same object, both algorithms (SMGBM and SMGAT) perform extremely well, yielding success rates of 97.4% and 99.5%, respectively.

4.2 Occluded Scenes

To illustrate the matching of an occluded image to the database, we compare an image containing the piggy bank occluded by the bulb socket, as shown in Figure 3. Table 3 shows the distance of the test image to the other images in Figure 2. The closest matching view is the middle view of the piggy back which is, in fact, the view embedded in the occluded scene. In a labeling task, the subgraph matching the closest model view would be removed from the graph and the procedure applied to the remaining subgraph. After removing the matching subgraph, we match the remaining scene subgraph to the entire database, as shown in Table 4. In this case, the closest view is the correct view (Figure 2(d)) of the socket.

Algorithm	2(a)	2(b)	2(c)	2(d)	2(e)
SMGBM	9.56	3.47	8.39	12.26	14.72
SMGAT	24.77	9.29	21.19	30.17	33.61

Table 3: Distance of Figure 3(a) to other images in Figure 2. The correct piggy bank view (Figure 2(b)) is the closest matching view.

5 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

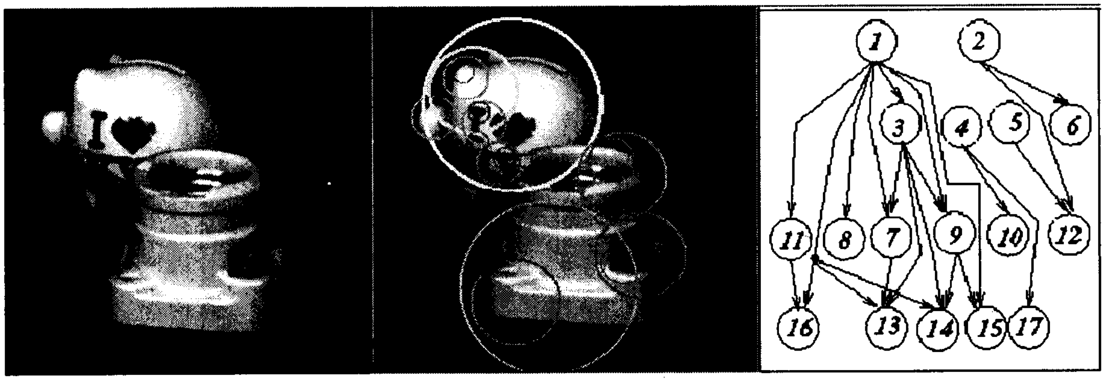


Figure 3: Occluded Object Matching: (a) original image; (b) saliency map; and (c) saliency map graph

Algorithm	2(a)	2(b)	2(c)	2(d)	2(e)
SMGBM	12.42	14.71	14.24	4.53	9.83
SMGAT	18.91	20.85	17.08	7.19	15.44

Table 4: Distance of Figure 3(a) (after removing from its SMG the subgraph corresponding to the matched piggy back image) to other images in Figure 2.

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

Table 5: 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.

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 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 5. For example, when leaving out 36 views, 91% of the 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.

5.1 Limitations

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 eval-

[3] The n views removed at step ℓ 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 ℓ.

uation 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.

6 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.

Acknowledgements

We gratefully acknowledge Columbia University's Shree Nayar for providing us with the database of model views. Sven Dickinson also gratefully acknowledges the support of the National Science Foundation (NSF CAREER IRI-9623913).

References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

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

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

[9] 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.

[10] 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.

[11] A. Shokoufandeh, I. Marsic, and S. Dickinson. View-based object recognition using saliency maps. Technical Report DCS-TR-339, Department of Computer Science, Rutgers University, New Brunswick, NJ 08903, August 1998.

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

[13] 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.