

# Recognizing Objects Using Color-Annotated Adjacency Graphs

Peter Tu, Tushar Saxena and Richard Hartley

GE - Corporate Research and Development,  
P.O. Box 8, Schenectady, NY, 12301.

T. Saxena : CMA Consulting Services, Schenectady, NY 12309

**Abstract.** We introduce a new algorithm for identifying objects in cluttered images, based on approximate subgraph matching. This algorithm is robust under moderate variations in the camera viewpoints. In other words, it is expected to recognize an object (whose model is derived from a template image) in a search image, even when the cameras of the template and search images are substantially different. The algorithm represents the objects in the template and search images by weighted adjacency graphs. Then the problem of recognizing the template object in the search image is reduced to the problem of approximately matching the template graph as a subgraph of the search image graph. The matching procedure is somewhat insensitive to minor graph variations, thus leading to a recognition algorithm which is robust with respect to camera variations.

## 1 Outline

The present paper describes a method for finding objects in images. The typical situation is that one has an image of the object sought – the template image. The task is to find the object in a new image, taken from a somewhat different viewpoint, possibly under different lighting. The method used is based on approximate attributed graph matching. As a first step, the image is segmented into regions of approximately constant color. The geometrical relationship of the segmented colored regions is represented by an attributed graph, in which each segment corresponds to a vertex in the graph, and proximate regions are joined by an edge. Vertices are annotated with the size, shape and color of the corresponding segment. Finding an object in a new image then comes down to an approximate graph-matching problem in which a match is sought in the new image for a subgraph approximating the one corresponding to the sought object. The graph matching can only be approximate, because of the inexactness of the segmentation process, and the changed aspect of the object, due to change of lighting, viewpoint, and possible partial occlusion.

There has been much previous work in the area of recognition from color. An important body of work is concerned with what has been broadly called color constancy [27, 12, 9, 11, 17, 18, 10, 4]. The concern of such papers is to recognize an object based on its color alone. Typically eigenspace or histogram techniques

or similar approaches are used to characterize an object. These methods rely on the distribution of colors in a usually vaguely defined region of an image. Under different conditions of lighting, the histogram, or eigenspace region or surface will vary. Various sophisticated models have been proposed for this changeability, ranging from very simple models such as simple intensity variability ([27]) to affine color transformations ([18]) and physical atmospheric illumination models ([19]). Generally such papers are not specifically concerned with locating an object to be recognized in an image, or in finding an object that occupies only a small part of an image. An exception is [19] in which recognition is at the level of individual pixels. In addition, any geometrical information about the relative locations of different colored parts of the image is usually lost (for instance in histogramming techniques).

Similar in concept, a recent popular approach to recognition has been the appearance-based learning method of Nayar and Murase, also Lin and Lee ([22, 25, 26, 21]). This approach relies uses surfaces in an eigenspace to represent the views of an object under different poses. The method becomes increasingly complex as the number of degrees of freedom of pose and lighting increase. Once more, such methods are best suited for recognition of an object that constitutes a complete image. Searching for a candidate object in a complex scene is treated as a separate issue.

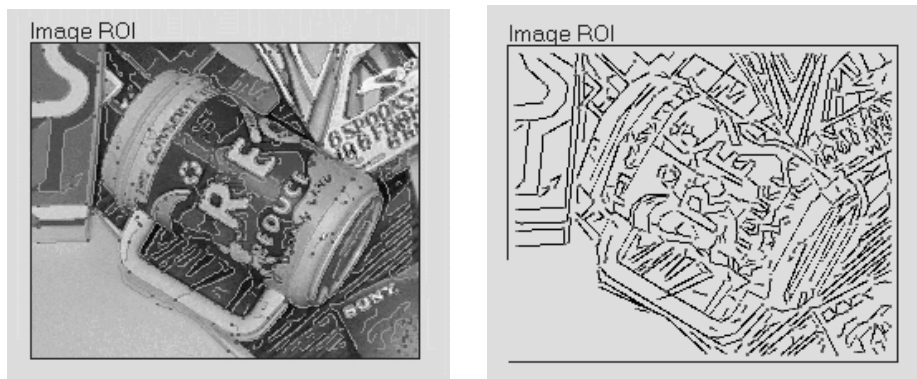
An alternative line of attack on object recognition has been to use the geometry of the object. Typically, this involves edge detection and grouping, followed by some sort of indexing or template matching based on geometry. Among many possible references we cite one ([32]). The present work seeks to amalgamate the color constancy and geometric approaches to object recognition. Previous work in amalgamating geometry and color includes [23, 5, 24, 7, 30, 6]. Earlier work of Hanson and Riseman ([15, 16]) lays a foundation for this approach. Among this cited work the approach of [5] is related to ours by dealing with blobs, which are ellipsoidal areas of consistent color. Similarly, in our approach, regions of an image segmented obtained from segmentation are represented by their principal moments, effectively treating them as ellipses.

## 2 Extracting Object Faces from Images

The first step in the algorithm is the division of the image into faces (or regions) of approximately constant color. The face extraction process proceeds in the three basic steps, which will be outlined in subsequent sections.

### 2.1 Detecting Approximate Region Boundaries

First the boundaries (that is edges) hypothesized to enclose the region are detected. As a first step in this process, the edges in the image are detected using a Canny-style edge detector, and line segments are fitted to the resulting edgels. It is reasonable to assume that the region boundaries pass through the resulting line segments, since under our assumptions, a face boundary will produce



**Fig. 1.** *Left* : The result of edge detection in a template image containing a cup. *Right* : The result of adjusting lines fitted to the cup edge segmentation.

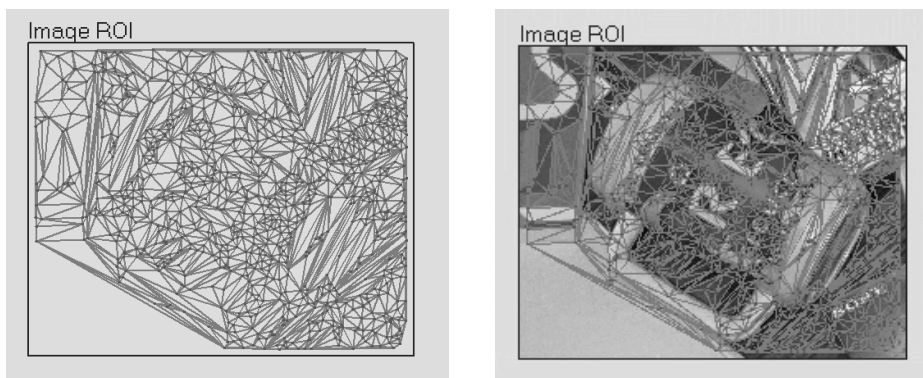
a discontinuity in the intensity and color variation of the enclosed region, and thus show up during edge detection. However, typically, these boundaries are detected in the form of numerous small broken line segments (see figure 1(left)). It is difficult to identify the exact geometry of the enclosed faces directly from these line segments. To improve the boundary geometries, we use some heuristics to further process the line segments. Some of these heuristics are:

- Merge lines that are nearly collinear and within some proximity threshold of each other. This is useful in creating a single edge which may have broken up into various smaller (but nearly parallel) segments during edge detection.
- Create T-Junctions from pairs of lines, one of which ends close to the inside of, and away from the endpoints of, the other. This is useful in recreating intersections of edges on occluding objects. This will help in obtaining well-defined faces on both objects.
- Intersect lines whose end-points are close to each other, and the lines are at obtuse angles. This recreates the corners of an object which may not have been detected during segmentation.

Figure 1(right) shows the result of applying these heuristics on the segmentation of the image in figure 1(left). In our experience, these heuristics aid significantly in correcting most of the degenerate boundary segments.

## 2.2 Estimating Initial Uniform Regions: Constrained Triangulation

Using the boundary line segments from the previous step, we now generate an initial partition of the image into triangles of uniform intensity and color. This is accomplished by a *constrained triangulation* of the boundary lines. A constrained triangulation produces a set of triangles which join nearest points (end-points of the lines), but respect the constraining boundary lines. That is, each boundary line segment will be an edge of some triangle.



**Fig. 2.** Constrained triangulation on the adjusted cup lines. On the left, the triangles only; on the right the triangles superimposed on the image.

Since all triangles are formed from the end-points and lines on the boundaries of the faces, each triangle lies completely inside a face. Moreover, since these triangles cover the whole image, each face can be represented by a union of a finite number of these triangles. As an example, see the result of constrained triangulation on an image segmentation in figure 2.

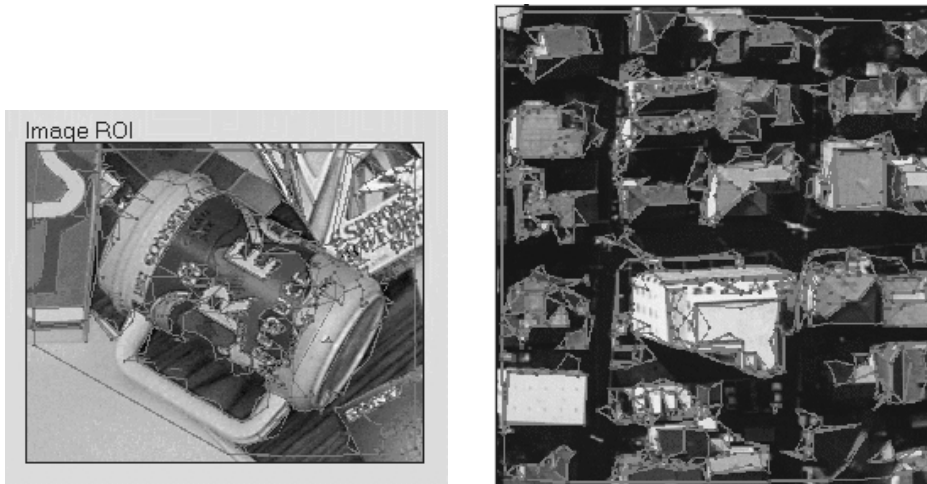
### 2.3 Extracting Object Faces: Region Merging

In the next step, a region-merging procedure is used to incrementally generate the visible object faces in the image. Starting with the triangular regions from the constrained triangulation, neighboring regions are successively merged if they have at least one of the following properties:

1. **Similar color intensities:** Two adjacent regions are merged if the difference between their average color intensity vectors is less than a threshold. This is a reasonable merging property since neighboring faces in objects are at angles to each other, and are likely to cast images of different intensities. As a refinement of this method, one could merge two regions based on a decision of which of two hypotheses (the two regions are separate; the two regions should form a single region) is preferable based on the color statistics of the regions. In addition, a linear or more complex color gradient over a face could be modeled. These methods have been suggested in [15, 16] but we have not tried them yet.
2. **Unsupported bridge:** Two adjacent regions are merged if the percentage of edges common between them, which are *unsupported*, is larger than a threshold. An edge is said to be supported if a specified percentage of its pixels belong to an edgel detected by the edge detector. Merging based on this property will ensure the inclusion of those boundary segments which were missing from the set of line segments derived from edgels. This is demonstrated by

the fact that a number of non-constraining lines in the triangulation end up being found supported.

After each merge, the properties (size and color) of the new, larger region are recomputed from the properties of the two regions being merged.



**Fig. 3.** *Faces of the cup (left) and urban scene (right) extracted by our algorithm.*

The merging iterations continue until the color intensities of each pair of neighboring regions are sufficiently different, and most of the edges common between them have support from the segmentation. Under our assumptions about the nature of the objects and the illumination, it is reasonable to assume that the resulting regions are likely to be images of the faces of objects pictured in the image. As an illustration, see figure 3 which shows the faces extracted using our algorithm.

The result of the segmentation and merging algorithm is a set of regions with associated color (RGB) values. Typically, there remain small narrow regions lying along region boundaries. These are removed from consideration, since they do not represent meaningful faces in the image, but are caused by color transition across a boundary. Similarly, any residual very small regions are removed. Thus, small regions less than about 30 square pixels sometimes remain after the region merging, since slight variations of color have prevented them from being merged with adjacent large regions.

### 3 Deriving Graph Representations of Objects

Once all the object faces in the image have been generated, they are represented as a graph. To capture the relative placements of the objects in the image,

and the topology of the scene, an adjacency graph of the faces in the scene is constructed.

Each vertex in the graph represents a region, and is annotated with the shape, position and color attributes of the region. Shape is represented by the moment matrix of the region, from which one may derive the area of the region, along with the orientation and ratio of the principal axes of the region. In effect, the region is being represented as an ellipse. This shape representation is of course an extremely rough representation of the shape of the region. However, it is also quite forgiving of variations of shape along the boundaries, or even a certain degree of fragmentation of the region. Since matching will not be done simply on the basis of a region-to-region match, but rather on matching of region clusters, this level of shape representation has proven to be adequate. More precise shape estimates have been considered, however. Their use must be dictated by the degree of accuracy and repeatability of the segmentation process, however. The color of the region is represented by an RGB color vector. Other representations are of course possible, and have been tried by other authors ([15, 16]).

Because of the possibility of regions being fragmented or regions being improperly merged, it turns out to be inappropriate to use edges in the graph to represent physically adjacent regions. The adjacency graph generated by such a rule is too sensitive to minor variations in the image segmentation. Instead the choice was made of joining each vertex to the vertices representing the  $N$  closest regions in the segmented image. A value of  $N = 8$  was chosen. Thus, each vertex in the graph has 8 neighbors.

#### 4 The three-tier matching method.

The reduction of the image to an attributed graph represents a significant simplification. The graph corresponding to a typical complicated image (the search image) may contain up to 500 or so vertices, whereas the graph corresponding to an object to be found (the template) may contain 50 vertices or so. Thus a complete one-on-one comparison may be carried out in quite a short time.

The search is carried out in three phases, as follows:

1. **Local comparison.** A one-to-one comparison of each pair of vertices is carried out. Each pair of vertices, one from the *template graph* and one from the *search graph* is assigned a score based on similarity of shape, size and color, within rather liberal bounds.
2. **Neighborhood comparison.** The local neighborhood consisting of a vertex and its neighbors in the template graph is compared with a local neighborhood in the search graph. A score is assigned to each such neighborhood pairing based on compatibility, and the individual vertex-pair scores.
3. **Global matching.** A complete graph-matching algorithm is carried out, in which promising matches identified in the stage-2 matching are pieced together to identify a partial (or optimally a complete) graph match.

Each of these steps will be described in more detail in later sections. The idea behind this multi-stage matching approach is to avoid ruling out possible

matches at an early stage, making the matching process robust to differences in the segmentation and viewpoint. This approach is motivated from the scoring method used in tennis matches in which a three-tier scoring system is used – game, set match. At each stage, slight advantages are amplified. A player who wins 55% of points will win 62% of games, 82% of sets and 95.7% of matches. Thus, the better player will (almost) always win despite temporary setbacks. In the same way the three-tier graph matching method provides a robust way of converging to the correct match, despite local fluctuations of region-to-region scoring.

#### 4.1 Local matching.

In local matching, individual vertex pairs are evaluated. Each pair is assigned a score based on shape and color. Recall, that each region is idealized as an ellipse. Shapes are compared on the basis of their size and eccentricity. Up to a factor of 2 difference in size is allowed without significant penalty. This allows for different scales in the two images, within reasonable bounds.

Because of different lighting conditions, colors may differ between two images. The most significant change in color, however is due to a brightness difference. To allow for this, colors are normalized before being compared. The color of a region is represented by a vector, and vectors that differ by a constant multiple are held to represent the same color.

The cost of a local match between two vertices is denoted by  $C_{\text{local}}$ .

#### 4.2 Neighborhood matching.

Each vertex (here called core) in the graph has eight neighbors representing the eight closest regions. In comparing the local neighborhood of one core vertex  $v_0$  with the local neighborhood of a potential match  $v'_0$ , an attempt is made to pair the neighbor nodes of  $v_0$  with those of  $v'_0$ . In this matching the order of the neighbor vertices must be preserved. Thus, let  $v_1, v_2, \dots, v_n$  be the neighbors of one core vertex, given in cyclic angular order around the core, and let  $v'_1, \dots, v'_m$  be the neighbors of a potential match core, similarly ordered. One seeks subsets  $S$  of the indices  $\{1, \dots, n\}$  and  $S'$  of the indices  $\{1, \dots, m\}$  and a one-to-one mapping  $\sigma : S \rightarrow S'$  so that the matching  $v_i \leftrightarrow v'_{\sigma(i)}$  preserves cyclic order. The total cost of a neighborhood match is equal to

$$C_{\text{nbhd}} = w_0 C_{\text{local}}(v_0, v'_0) + \sum_{i \in S} w_i C_{\text{local}}(v_i, v'_{\sigma(i)})$$

where  $w_i$  is a weight between 0 and 1 that depends on the ratio of distances between the core vertices and the neighbors  $v_i$  and  $v'_{\sigma(i)}$ . For each pair of core vertices  $v_0, v'_0$ , the neighborhood matching that maximizes this cost function is speedily and efficiently found by dynamic programming.

### 4.3 Graph matching

In previous sections, the template image and the search image were reduced to a graph, and candidate matches between vertices in the two images were found. The goal of this section is to generate a mutually consistent set of vertex matches between the template and the search image. An association graph  $\mathbf{G}$  [2, 8] provides a convenient framework for this process. In considering the association graph, it is important not to confuse it with the region adjacency graph that has been considered so far. In the association graph, vertices represent pairs of regions, one from each image. Such a vertex represents a hypothesized matching of a region from the template image with a region from the search image. Weighted edges in the association graph represent compatibilities between the region matchings denoted by the two vertices connected by the edge.

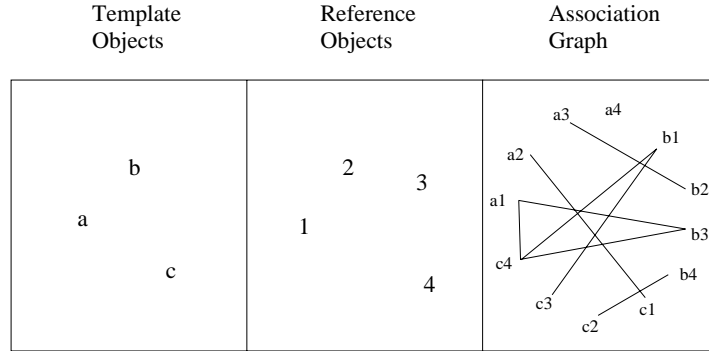
Thus, a vertex in the association graph is given a double index, and denoted  $v_{ij}$ , meaning that it represents a match between region  $R_i$  in the template image and region  $R'_j$  in the search image. This match may be denoted by  $R_i \leftrightarrow R'_j$ . As an example, if  $j_1 \neq j_2$  then  $v_{ij_1}$  is not compatible with  $v_{ij_2}$ . This is because vertex  $v_{ij}$  represents a match  $R_i \leftrightarrow R'_{j_1}$  and  $v_{ij_2}$  represents the match  $R_i \leftrightarrow R'_{j_2}$ , and it is impossible that region  $R_i$  should match both  $R'_{j_1}$  and  $R'_{j_2}$ . Thus, vertices  $v_{ij_1}$  and  $v_{ij_2}$  are incompatible and there is no edge joining these two vertices in the association graph. There are other cases in which matches are incompatible. For instance, consider a vertex  $v_{ij}$  representing a match  $R_i \leftrightarrow R'_j$  and a vertex  $v_{kl}$  representing a match  $R_k \leftrightarrow R'_l$ . If regions  $R_i$  and  $R_k$  are close together in the template image, whereas  $R'_j$  and  $R'_l$  are far apart in the search image, then the matches  $R_i \leftrightarrow R'_j$  and  $R_k \leftrightarrow R'_l$  are incompatible, and so there is no edge joining the vertices  $v_{kl}$  and  $v_{ij}$ . Matches may also be incompatible on the grounds of orientation or color.

Formally, the association graph  $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$  is composed of a set of vertices  $\mathbf{V}$  and a set of weighted edges  $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$ . Each vertex  $v$  represents a possible match between a template region and a search region. If there are  $N$  template regions and  $M$  search regions then  $\mathbf{V}$  would have  $NM$  vertices (see figure 4). In order to reduce the complexity of the problem, the graph  $\mathbf{G}$  is pruned so that only the top 5 assignments for each template region are included in  $\mathbf{V}$ . These nodes are labeled  $v_{ij}$  which is interpreted as the  $j$ th possible assignment for the  $i$ th template region. A slack node for each template region is inserted into the graph. The slack node  $v_{i0}$  represents the possibility of the NULL assignment for the  $i$ th template region, that is, no matching region exists in the other image. If an edge  $e = (v_{ij}, v_{kl})$  exists then the assignments between nodes  $v_{ij}$  and  $v_{kl}$  are considered compatible. The weights for the edges are derived from the compatibility matrix  $\mathbf{C}$  which is defined as:

$$C_{(ij)(kl)} = \begin{cases} 0 & \text{if } j = 0 \text{ or } l = 0 \\ 0 & \text{if } i = k \text{ and } j \neq l \\ 0 \text{ to } 1 & \text{if } (i, j) = (k, l) \\ 0 \text{ to } 1 & \text{if } v_{ij} \text{ and } v_{kl} \text{ are compatible} \\ -N & \text{if vertices } v_{ij} \text{ and } v_{kl} \text{ are not compatible} \end{cases}$$



Where  $N$  is the number of template regions. The value of  $C_{(ij)(ij)}$  represents the score given to the individual assignment defined by node  $v_{ij}$ . A subgraph of  $\mathbf{G}$  represents a solution to the matching problem. The choice of weight  $N$  for an incompatible match is to discriminate against incompatible matches and make certain that a set of edges with maximum weight represents a clique of compatible matches.



**Fig. 4.** The template and search images are reduced to a set of regions. Each possible pair of assignments are assigned to a node in the association graph. Edges in the graph connect compatible assignments.

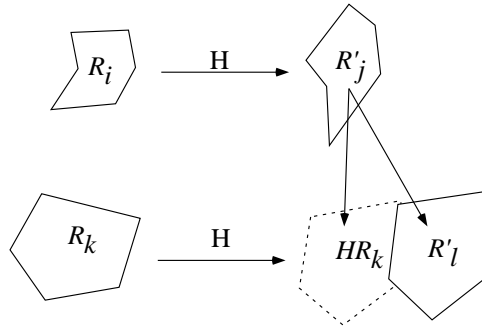
The method of determining compatibility and assigning compatibility scores  $C_{(ij)(kl)}$  for compatible matches is as follows. Consider a candidate region pair  $R_i \leftrightarrow R'_j$ . The local neighborhood of region  $R_i$  has been matched with neighborhood  $R'_j$  during the neighborhood matching stage. In doing this, a set of neighbors of the region  $R_i$  have been matched with the neighbors of the region  $R'_j$ . This matching may be considered as a correspondence of several regions (a subset of the neighbors of  $R_i$ ) with an equal number of regions in the other graph. From these correspondences a projective transformation is computed that maps the centroid of  $R_i$  to the centroid of  $R'_j$  while at the same time as nearly as possible mapping the neighboring regions of  $R_i$  to their paired neighbors of  $R'_j$ . Thus, the neighborhood correspondence is modeled as closely as possible by a projective transformation of the image. Let  $H$  be the projective transformation so computed.

Now let  $R_k \leftrightarrow R'_l$  be another candidate region match. To see how well this is compatible with the match  $R_i \leftrightarrow R'_j$ , the projective transformation  $H$  is applied to the region  $R_k$  to see how well  $H(R_k)$  corresponds with  $R'_l$ . As a measure of this correspondence, the vector from  $R'_j R'_l$  is compared with the vector  $R'_j H(R_k)$ . This is illustrated in figure 5. A compatibility score is assigned based on the angle and length difference between these two vectors. The two assignments are deemed incompatible if the angle between the two vectors exceeds 45 degrees, or their length ratio exceeds 2.

A color compatibility score is also defined. The correspondence of a core vertex and its neighbors with the matched configuration in the other image can be used to define an affine transformation of color space from the one image to the other. An affine color transformation is a suitable model for color variability under different lighting conditions ([18]). The affine transformation defined for one matched node pair is used to determine whether another matched node pair is compatible.

The final compatibility score is computed as

$$C_{(ij)(kl)} = C_{\text{nbhd}}(i, j) \times C_{\text{nbhd}}(k, l) \times \text{Angle compatibility score} \times \text{length ratio compatibility score} \times \text{color compatibility score}$$



**Fig. 5.** Compatibility of two matches is determined by applying the transformation  $H$  defined by the neighbors of the first pair  $(R_i, R'_j)$  to the region  $R_k$  belonging to the second pair. The positions of  $HR_k$  and  $R'_l$  relative to  $R'_j$  are compared.

#### 4.4 Solution Criteria

The Hough transform or matched filtering approach assumes that a global transformation defined by a relatively small set of parameters can be used to map the template regions onto the search regions. The largest set of nodes in  $\mathbf{V}$  which is consistent with a particular transformation would then constitute a final solution. However just because two nodes are consistent with a particular transformation does not necessarily imply that the two nodes are consistent with each other. For instance, in the association graph of 4, a match  $(c, 4)$  is compatible with  $(b, 1)$  and  $(b, 1)$  is compatible with  $(c, 3)$ . However  $(c, 3)$  is not compatible with  $(c, 4)$ , since  $c$  can not be simultaneously matched with both 3 and 4.

A popular graphical approach which can take advantage of some of the information contained in the edge structure is a node clustering technique where a simple depth first search is used to determine the largest connected subgraph of

**G.** A connected graph is one in which a path of edges exist between every pair of nodes in the graph. This solution represents a certain amount of consistency. However as before, the statement that node  $a$  is consistent with node  $b$  and node  $b$  is consistent with node  $c$  does not necessarily imply that node  $a$  is consistent with node  $c$ . This leads to the conclusion that in order to take full advantage of the mutual constraints embedded in the association graph, the final solution should represent a clique on **G**.

A subset  $\mathbf{R} \subseteq \mathbf{V}$  is a clique on **G** if  $v_{ij}, v_{kl} \in \mathbf{R}$  implies that  $(v_{ij}, v_{kl}) \in \mathbf{E}$ . The search for a maximum clique is known to be an *NP* complete problem [14]. Even after pruning, the computational costs associated with exhaustive techniques such as [1] would be prohibitive. It has been reported [3] that determining a maximum clique is analogous to finding the global maximum of a binary quadratic function. Authors such as [20, 28] have taken advantage of this idea by using relaxation and neural network methods to approximate the global maximum of a quadratic function, where this maximum corresponds to the largest clique in the association graph. Although the largest clique, which is based on the information contained in **E**, ensures a high level of mutual consistency, the nuances of the compatibility measures in **C** are lost. In order to take advantage of the continuous nature of these edge strengths, a quadratic formula is specified where the global maximum corresponds to the clique that has the maximum sum of internal edge strengths. An approach based on Gold and Rangarjans’s gradual assignment algorithm (GAA) is used to estimate the optimal solution. The GAA is an iterative optimization algorithm which treats the problem as a continuous process but converges to a discrete solution. Even though the solution might be generated based on a local maximum this solution will be guaranteed to be a maximal clique. A *maximal* clique is one that is not a proper subset of any other clique.

#### 4.5 Binary Quadratic Formulation

A binary solution column vector  $\mathbf{m}$  is defined such that if  $m_{ij} = 1$  then  $v_{ij}$  is part of the final solution and if  $m_{ij} = 0$  then  $v_{ij}$  is excluded from the final solution. If the slack node  $v_{i0}$  is part of the final solution then the template region  $i$  has no assignment. The columns and rows corresponding to the slack nodes in the compatibility matrix are filled with 0 entries. From a graph theory point of view, the slack nodes are connected to all other nodes by edges with zero weight.

The binary quadratic formula  $F(\mathbf{m})$  is defined as:

$$F(\mathbf{m}) = \mathbf{m}^T \mathbf{C} \mathbf{m} \quad (1)$$

where **C** is the compatibility matrix defined in section 4.3. In order to ensure that each template region can be mapped to at most one search region, the final solution is constrained such that

$$\sum_{j=1}^6 m_{ij} = 1 \quad \text{for all } i \text{ .} \quad (2)$$

A solution corresponding to a global maximum of  $F(\mathbf{m})$  represents a set of assignments with the largest amount of mutual compatibility. Any maximum of  $F(\mathbf{m})$  (global or local) represents a maximal clique on  $G$ . To show this consider a particular solution  $\hat{\mathbf{m}}$  where there exists  $i, j, k, l$  such that  $\hat{m}_{ij} = 1$  and  $\hat{m}_{kl} = 1$  but that the nodes  $v_{ij}$  and  $v_{kl}$  are incompatible assignments. Clearly this is the only condition necessary for the solution  $\hat{\mathbf{m}}$  not to qualify as a clique. A second solution  $\bar{\mathbf{m}}$  is introduced, the same as  $\hat{\mathbf{m}}$  except that  $\bar{m}_{ij} = 0$  and  $\bar{m}_{i0} = 1$ , which means that region  $i$  has no assignment. Using the definition of  $\mathbf{C}$  (see section 4.3) and equations 1 and 2 it can be shown that the difference between  $F(\bar{\mathbf{m}})$  and  $F(\hat{\mathbf{m}})$  is

$$F(\bar{\mathbf{m}}) - F(\hat{\mathbf{m}}) = 0 - 2 \sum_{q=1}^N \sum_{r=1}^6 \hat{m}_{qr} C_{(ij)(qr)} \geq 2(N - (N - 1)) = 2 \quad (3)$$

Therefore  $F(\hat{\mathbf{m}})$  does not represent a maximum which means that only a clique on  $G$  can generate a maximum on  $F(\mathbf{m})$ . The next step is to find a solution which is a maximum of  $F(\mathbf{m})$ .

#### 4.6 Approximating the Clique with the largest degree of mutual compatibility

As previously stated the search for global maximum of 0-1 quadratic equations is known to be an  $NP$  complete problem so that an approximate solution to the optimum value of  $F(\mathbf{m})$  will have to be estimated. The GAA is a recursive routine used to solve a general assignment problem under the constraints that assignments must be one to one. Any binary quadratic cost function can be used to drive the GAA optimization process. When generating the compatibility matrix, two nodes  $v_{ij}$  and  $v_{kl}$  are considered incompatible if they map template regions  $i$  and  $k$  to the same search region. Inclusion of  $v_{ij}$  and  $v_{kl}$  in the final solution would contradict the statement that a final solution is guaranteed to be a maximal clique. This means that the portion of the GAA that prevents a many to one condition from occurring need not be implemented.

Initially  $\mathbf{m}$  is treated as a continuous vector. Several constraints are placed on the optimization process:

$$\forall ij \quad m_{ij} \geq 0 \quad (4)$$

$$\forall i \quad \sum_{j=1}^6 m_{ij} = 1. \quad (5)$$

During each iteration  $t$  the update rule for the GAA is as follows:

$$m_{ij}(t+1) = \frac{e^{\beta \frac{\delta F(t)}{\delta m_{ij}(t)}}}{\sum_k^6 e^{\beta \frac{\delta F(t)}{\delta m_{ik}(t)}}} \quad (6)$$

where  $\beta$  is a positive number, and

$$\frac{\delta F(t)}{\delta m_{ij}(t)} = 2 \sum_{p=1}^N \sum_{q=1}^6 m_{pq}(t) C_{(ij)(pq)}. \quad (7)$$

The update equation 6 ensures that conditions 4 and 5 are maintained. Initially  $\beta$  is set to a low value so that multiple solutions can coexist. The value of  $\beta$  is gradually increased. As can be seen from equation 6, as  $\beta$  becomes large the values of  $\mathbf{m}$  are forced to discrete values of 0 or 1.

Figure 6 shows an example of the optimization process. A sequence of snapshots graphically displays the evolution of the solution vector for a template image of 15 regions. After the first initial iterations, the NULL assignments are favored because of the inconsistencies between rival solutions. Between time 1 and time 3 a dominant solution begins to emerge. The solution is refined during time 4 and time 5. At time 6 the algorithm has converged to a final solution and by time 7 the coefficients have taken on binary values.

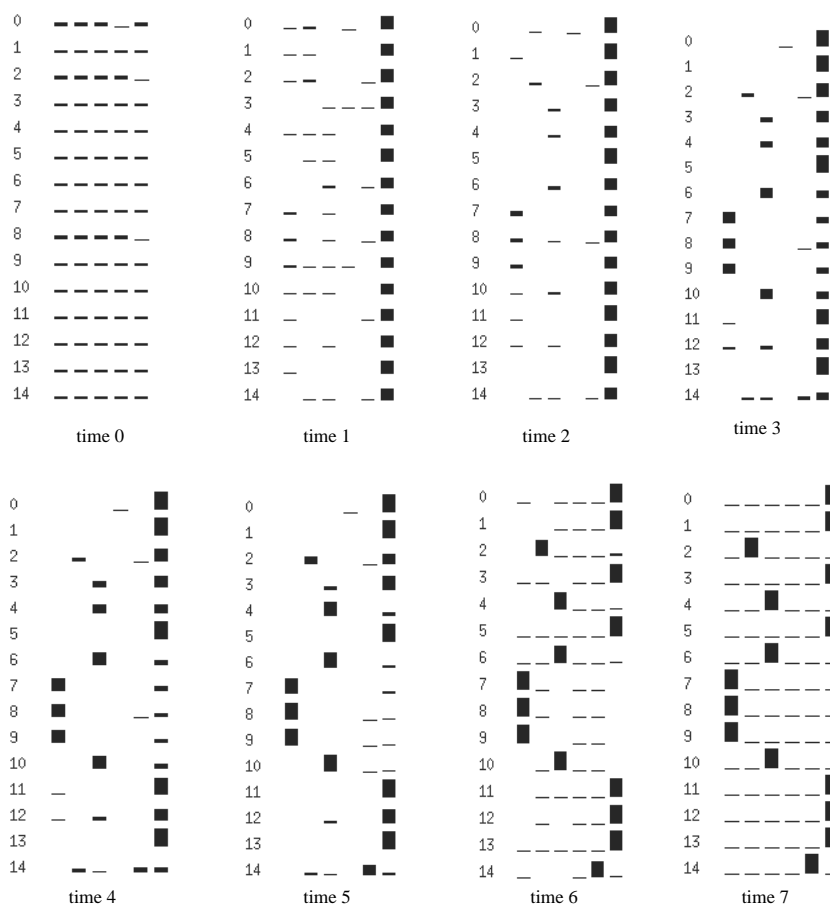
## 5 Results

The algorithm was tried on several sets of color images. The first example was a computer manual, shown in figure 7. The manual was easily found in different images of a cluttered table-top, even when the manual was partially occluded. Note that a second manual shown in the images is not found, since it is actually a different color, though this is not obvious from the grey-scale images shown in the paper. Other examples are shown in figures 10 and 9.

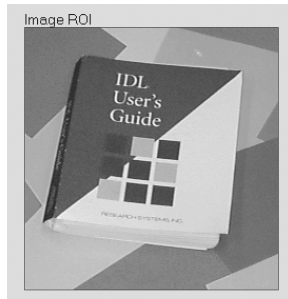
## 6 Conclusion.

The amalgamation of region segmentation algorithms with modern color constancy methods gives the possibility of improved object recognition in color and multi-spectral imagery. The adoption of an inexact graph-matching approach makes recognition independent of moderate lighting and view-point changes. The graph matching approach was able to generate solutions with consistency at multiple levels. The region adjacency graphs were able to highlight image to template correspondences with strong local support. By insisting that the final solution must represent a clique on the association graph, global consistency was achieved. Although the maximum clique problem is NP complete, it was demonstrated that strong maximal cliques can be generated using a variation of the gradual assignment algorithm.

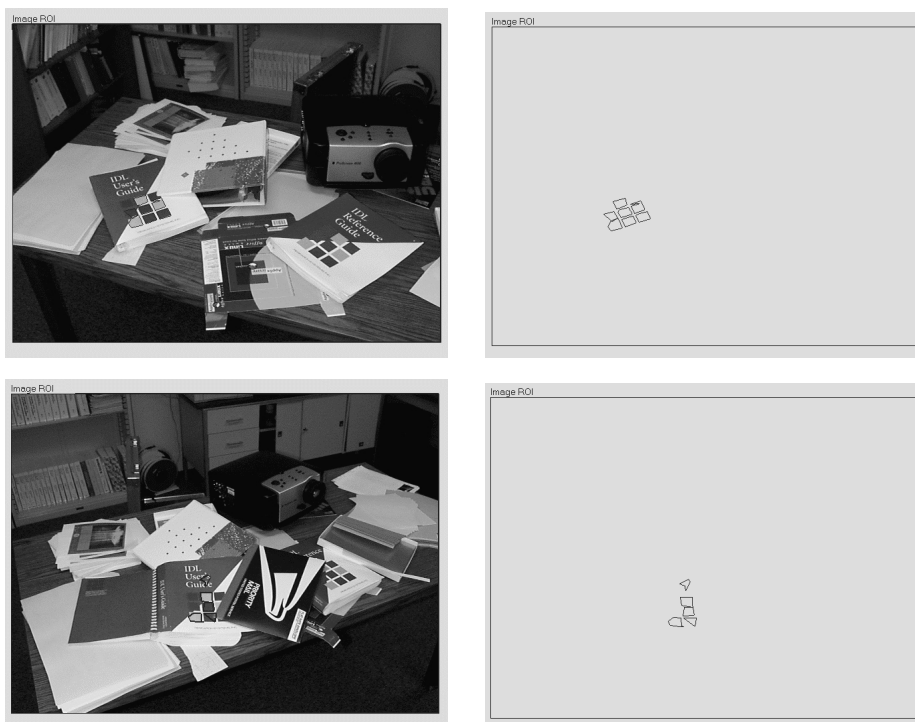
## Evolution of Decision Vector



**Fig. 6.** Illustration of the GAA optimization process. The coefficients for the solution vector  $\mathbf{m}$  are shown at various points in time. Each row represents the coefficients corresponding to a particular template region. The last column at each time represent the coefficients for the NULL assignments. Initially the coefficients take on continuous values between 0 and 1. By the end of the process only binary values exist.



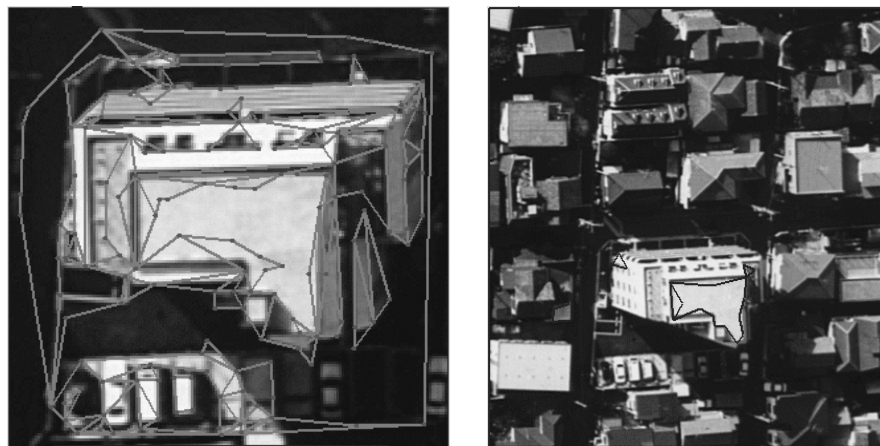
**Fig. 7.** *The computer manual used as a template*



**Fig. 8.** *Two examples of recognition. On the left the search image, and on the right the outlines of the regions matched against the template.*



**Fig. 9.** Recognition of cup image. On the left is the template, in the center the search image and on the right the identified regions of the located cup. Note that the cup in the search image is seen from a different angle from the template image. The letters REC are visible in the template, but only RE is visible in the search image.



**Fig. 10.** Recognizing a building. On the left the template, and on the right the search image showing the recognized building.



## References

1. Ambler A.P., Barrow H.G., Brown C.M., Burstall R.M., Popplestone R.J., 'A versatile computer-controlled assembly system', IJCAI, pages 298-307, (1973).
2. Ballard D.H., Brown C.M., 'Computer Vision', Prentice-Hall, Englewood Cliffs, NJ, (1982).
3. Batahan F., Junger M., Reinelt G., 'Experiments in Quadratic 0-1 programming', Mathematical Programming, vol. 44, pages 127-137, (1989).
4. David H. Brainard and Brian A. Wandell, 'Analysis of the retinex theory of color vision', Journal of the Optical Society of America, Vol. 3, No 10, pages 1651 – 1661, (1986).
5. J. Brian Burns and Stanley J. Rosenschein, 'Recognition via Blob Representation and Relational Voting', Proc 27th Asilomar Conference on Signals, Systems and Computer, pages 101 – 105, (1993).
6. Marie-Pierre Dubuisson and Anil K. Jain, 'Fusing Color and Edge Information for Object Matching', Proceedings, ICIP-94, pages 471 – 476, (1994).
7. Francois Ennesser and Gerard Medioni, 'Finding Waldo, or Focus of Attention using Local Color Information', IEEE Transactions on PAMI, Vol 17, 8, pages 805–809, (1993).
8. Faugeras O., 'Three-dimensional computer vision', MIT Press, (1993).
9. G. D. Finlayson, B. V. Funt and K. Barnard, 'Color Constancy under Varying Illumination', Proceedings of 5th International Conference on Computer Vision, ICCV-95, pages 720 – 725, (1995).
10. David Forsyth, 'A Novel Approach to Colour Constancy', Proceedings of 2nd International Conference on Computer Vision, ICCV-88, pages 9 – 18, (1988).
11. Brian V. Funt and Graham D. Finlayson, 'Color Constant Color Indexing', IEEE Transactions on PAMI, Vol. 17, No. 5, pages 522–529, (May 1995).
12. Graham D. Finlayson, Mark S. Drew and Brian V. Funt, 'Color constancy: generalized diagonal transforms suffice', Journal of the Optical Society of America, Vol. 11, No 11, pages 3011–3019, (1994).
13. Gold S. and Rangarjan A., 'A gradual assignment algorithm for graph matching', IEEE Transactions on PAMI, Vol. 18 No 4, (April 1996), pages 377 – 387.
14. Gibson A., 'Algorithmic graph theory', Cambridge University Press, Cambridge (MA), USA, (1985)
15. Allen R. Hanson and Edward M. Riseman, 'Segmentation of Natural Scenes', in Computer Vision Systems, (edited A. Hanson and E. Riseman), Academic Press, (1978), pages 129 – 164.
16. Allen R. Hanson and Edward M. Riseman, 'VISIONS : A computer system for interpreting scenes', in Computer Vision Systems, (edited A. Hanson and E. Riseman), Academic Press, (1978), pages 303 – 334.
17. Glenn Healey and David Slater, 'Global color constancy : recognition of objects by use of illumination-invariant properties of color distributions', Journal of the Optical Society of America, Vol. 11, No 11, pages 3003 – 3010, (1994).
18. Glenn Healey and David Slater, 'Computing Illumination-Invariant Descriptors of Spatially Filtered Color Image Regions', IEEE Transactions on Image Processing, Vol. 6 No 7, (July 1997), pages 1002 – 1013.
19. Glenn Healey and David Slater, 'Exploiting an Atmospheric Model for Automated Invariant Material Identification in Hyperspectral Imagery', Preprint report : to appear (Darpa IU Workshop, Monterey, (1998) ?).

20. Lin, F. 'A parallel computation network for the maximum clique problem', Proceeding 1993 international symposium on circuits and systems, pages 2549-52, vol. 4, IEEE, (May 1993).
21. Stephen Lin and Sang Wook Lee, 'Using Chromaticity Distributions and Eigenspace Analysis for Pose, Illumination and Specularity Invariant Recognition of 3D objects', Proceedings Computer Vision and Pattern Recognition, CVPR-97, pages 426 – 431, (1997).
22. Hiroshi Murase and Shree K. Nayar, 'Visual Learning and Recognition of 3-D Objects from Appearance', International Journal of Computer Vision, 14, pages 5–24, (1995)
23. Adnan A. Y. Mustafa, Linda G. Shapiro and Mark A. Ganter, '3D Object Recognition from Color Intensity Images', Proc. ICPR'96, pages 627 – 631, (1996).
24. Kenji Nagao, 'Recognizing 3D Objects Using Photogrametric Invariant', Proceedings of 5th International Conference on Computer Vision, ICCV-95, pages 480 – 487, (1995).
25. Shree K. Nayar, Sameer A. Nene and Hiroshi Murase, 'Real-Time 100 Object Recognition System', Proc. 1996 IEEE Conference on Robotics and Automation, Minneapolis, pages 2321 – 2325, (April 1996).
26. Sameer A. Nene and Shree K. Nayar, 'A Simple Algorithm for Nearest Neighbor Search in High Dimensions', IEEE Transactions on PAMI, Vol. 19 No 9, pages 989–1003, (Sept, 1997).
27. Michael J. Swain and Dana H. Ballard, 'Color Indexing', International Journal of Computer Vision, 7:1 pages 11-32, (1991).
28. Pelillo M., 'Relaxation labeling Networks that solve the maximum clique problem', Fourth international conference on artificial neural networks, pages 166-70, published by IEE (June 1995).
29. Tushar Saxena, Peter Tu and Richard Hartley, 'Recognizing objects in cluttered images using subgraph isomorphism', to appear in Proceedings of the IU Workshop, Monterey, (1998).
30. David Slater and Glenn Healey, 'Combining Color and Geometric Information for the Illumination Invariant Recognition of 3D Objects', Proceedings of 5th International Conference on Computer Vision, ICCV-95, pages 563 – 568, (1995).
31. David Slater and Glenn Healey, 'Exploiting an Atmospheric Model for Automated Invariant Material Identification in Hyperspectral Imagery', Preprint report : to appear (Darpa IU Workshop, Monterey, (1998) ?).
32. A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu, N. Pillow, '3D Object Recognition Using Invariance', Artificial Intelligence Journal, 78, pages 239–288, (1995).