Object-Based Image Labeling through Learning-by-Example and Multi-Level Segmentation

Y. Xu [(1)], P. Duygulu [(2)], E. Saber [(1), (3)], A. M. Tekalp [(1), (4)], and F. T. Yarman-Vural [(2)]

(1). Department of Electrical and Computer Engineering, University of Rochester,

Rochester, NY 14627, USA

E-mail: {yaxu, saber, tekalp}@ece.rochester.edu

(2). Department of Computer Engineering, Middle East Technical University, Ankara, Turkey

E-mail: {duygulu, vural}@ceng.metu.edu.tr

(3). Xerox Corporation, 800 Phillips Road, Webster, NY 14580, USA

E-mail: eli.saber@usa.xerox.com

(4). College of Engineering, Koc University, Sariyer, Istanbul, TURKEY

E-mail: mtekalp@ku.edu.tr

Corresponding Author:

A Murat Tekalp, Distinguished Professor

Department of Electrical and Computer Engineering, Hopeman 204,

University of Rochester, Rochester, NY 14627-0126

Phone: (716) 275-3774

FAX: (716) 273-4919

E-mail: **tekalp@ece.rochester.edu**

SUMMARY

Most existing multimedia search and browsing systems allow users to utilize manually annotated keywords (at the semantic level) and/or low-level visual features for annotation and indexing. Great challenges still remain in automatically building connections between low-level features and semantic attributes, such as objects, people, and places. In order to narrow that gap, this paper proposes a method for automatic extraction and labeling of meaningful image objects using "learning by example" and threshold-free multi-level image segmentation. The proposed approach scans through images, each of which is pre-segmented into a hierarchical uniformity tree, to seek and label objects that are similar to an example object presented by the user. By representing images with stacks of multi-level segmentation maps, objects can be extracted in the segmentation map level with adequate detail. The proposed method provides an effective approach for automatic content analysis for object based image description and labeling. Experiments have shown that the proposed multi-level image segmentation results in significant reduction in computational complexity for object extraction and labeling (compared to single fine-level segmentation map) by avoiding unnecessary tests of combinations in finer levels. The multi-level segmentation based approach also achieves better accuracy in detection and labeling of small objects.

Object-Based Image Labeling through Learning-by-Example and Multi-Level Segmentation

Y. Xu [(1)], P. Duygulu [(2)], E. Saber [(1), (3)], A. M. Tekalp [(1), (4)], and F. T. Yarman-Vural [(2)]

(1). Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627, USA

E-mail: {yaxu, saber, tekalp}@ece.rochester.edu

(2). Department of Computer Engineering, Middle East Technical University, Ankara, Turkey

E-mail: {duygulu, vural}@ceng.metu.edu.tr

(3). Xerox Corporation, 800 Phillips Road, Webster, NY 14580, USA

E-mail: eli.saber@usa.xerox.com

(4). College of Engineering, Koc University, Sariyer, Istanbul, TURKEY

E-mail: mtekalp@ku.edu.tr

ABSTRACT

We propose a method for automatic extraction and labeling of semantically meaningful image objects using "learning by example" and threshold-free multi-level image segmentation. The proposed method scans through images, each of which is pre-segmented into a hierarchical uniformity tree, to seek and label objects that are similar to an example object presented by the user. By representing images with stacks of multi-level segmentation maps, objects can be extracted in the segmentation map level with adequate detail. Experiments have shown that the proposed multi-level image segmentation results in significant reduction in computation complexity for object extraction and labeling (compared to a single fine-level segmentation) by avoiding unnecessary tests of combinations in finer levels. The multi-level segmentation based approach also achieves better accuracy in detection and labeling of small objects.

Keywords: ***Object-based image labeling, Multi-level segmentation, Hierarchical content description, Learning by example.***

1 INTRODUCTION

Early multimedia search, browsing and retrieval systems employed text based image annotation and indexing. Two major difficulties were encountered with text based approaches: First, manual annotation, a necessary step for these approaches, is labor intensive and becomes impractical when the multimedia collection is large. Second, key word annotations are subjective, the same image/video may be annotated differently by different human observers.

To overcome these difficulties, multimedia applications started to allow users to utilize low-level visual features, such as color, texture, shape, and motion for indexing (see [1, 2, 3, 4] for recent surveys). Examples of such systems include: Trademark system [5]; QBIC [6]; Photobook [7]; VisualSEEK [8]; WebSEEK [9]; Chabot [10]; Illustra/Virage [11]; FourEyes [12]; MARS [13]; RetrievalWare (from Excalibur Technologies) [14]. Recent work also addressed integration of semantics into such systems using relevance feedback [25, 26, 27, 28]. However, great challenges still remain in bridging the gap between low-level image features and semantic attributes, such as objects, people, and places [1, 29]. While these concepts come naturally to human observers, they pose a significant challenge to automatic information processing systems. As pointed out by [30], virtually all the proposed systems utilize only low-level representations that have limited semantics. In order to narrow down this so-called "semantic gap" between the low level features and semantic abstraction, object-based content analysis, which performs semantically meaningful object segmentation on images, is an essential step.

Although there are a large number of image segmentation techniques available in the literature [15], semantically meaningful and accurate visual object segmentation remains as a difficult task. Without additional source of knowledge, automatic image segmentation based on low-level image features is unlikely to succeed in extracting semantic objects in generic images. From the perspective of image processing, automatic image segmentation computes local regions that are "homogenous" with respect to one or more low-level features, e.g., color, texture, according to some similarity measure. There are two problems with these methods: i) Homogeneity of low-level feature values may not correlate well with semantic meaning. For example, a semantic object may contain multiple colors and/or multiple motions. ii) The degree of homogeneity of a region is generally quantified by one or more threshold values for a given measure [16, 17, 18, 19]. However, selection of appropriate threshold values for the desired degree of homogeneity is generally application dependent. Therefore, it is difficult to design generic segmentation schemes that can extract semantic objects from images.

Recognizing the importance of bridging "the semantic gap" in object-based image labeling, this paper proposes the following novelties:

- Hierarchical content labeling and learning by example: We present in Section 2 a hierarchical content representation and the concept of learning by example based on color and shape cues that we initially proposed in [20] to automatically extract semantically meaningful visual objects. Hierarchical content trees are used to represent objects as composite nodes in the higher levels of the content trees, where image regions with uniform color or texture form the lowest level leaf nodes.

- Threshold-free multi-level segmentation**:** Rather than trying to achieve a single low-level segmentation map based on certain threshold values, in Section 3 we propose a threshold-free multi-level segmentation method to generate a stack of segmentations, which are ordered in a tree, called a *uniformity tree*. The closer we approach the top layer, the less the number of regions (which represent coarser detail) in the segmentation map.
- Optimized top-down search (for labeling) within a uniformity tree: Section 4 addresses some of the problems encountered in the implementation of the proposed learning by example method with a single layer segmentation, and proposes an optimized top-down search strategy based on multi-level segmentation to solve these problems.

Section 5 demonstrates the proposed concepts and provides a performance comparison between single and multi-level segmentation schemes. Conclusions are drawn in Section 6.

2 HIERARCHICAL CONTENT LABELING AND LEARNING BY EXAMPLE

We propose a region-based hierarchical image content description represented by a "scene graph" which consists of an *object-region tree* and an *adjacency matrix* [20, 21]. The *object-region tree* indicates the containment relationships between semantically meaningful objects, and image regions with uniform color or texture or other low-level image features. The *adjacency matrix* [20, 21] captures the spatial relationships between these low-level image regions. In this section, we first define such a hierarchical content representation that captures semantically meaningful object-based content structures in an image. We then present a procedure for automatic generation of the *object-region* tree through learning by example.

2.1 Hierarchical Content Representation

Let N denote the number of regions in the low-level image segmentation map, and L represent the number of levels within the *object-region* tree. The root node I of the tree, at level $l = 1$, corresponds to the whole image. Each leaf node $R_i^p(l)$ (also called elementary node) represents a homogeneous image region with uniform color or texture as indicated by the image segmentation map, where p, l and i denote the parent node index, the level of the tree $\{l = 1...L\}$, and the leaf node index $\{i = 1...N\}$ within level l, respectively. Furthermore, an intermediate node (also called composite node), denoted by $O_k^p(l)$ represents a semantic object, where $k = 1,..., M_l$ is the index for objects in the l^{th} level and M_l denotes the number of nodes within level l. For example, the root node I can also be represented as $O_{k=1}^{p=0}(1)$ indicating that it is at level $l=1$, it has no parent $(p=0)$, and it is the first node within that level $(k=1)$. On the other hand, the adjacency matrix $\mathbf{A}=\{a_{ij}\}$ is such that

$$a_{ij} = \begin{cases} 1 & \text{if } R_i^p(l) \text{ is a neighbor of } R_j^p(l) \ \forall i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

The hierarchical *object-region* tree is illustrated in Figure 1. Figure 1b displays a low-level color segmentation map for the image shown in Figure 1a. In Figure 1b, there are *N=8* separate regions, which are represented by elementary nodes in Figure 1d. The node I (also noted as $O_1^0(1)$) represents the parent node, as indicated by *p=1,* for the elementary nodes $R_i^{p=1}(2)$ *{i=1…N}* at level *l=2* in the content tree (see Figure 1d). Figure 1c shows the *adjacency matrix* for the segmentation map in Figure 1b. Figure 1e shows a "car" composite node $O_1^1(2)$ in the parent-child relationship tree. Note that this composite node has the root node as its parent, and the elementary nodes i=1 through 7 as its children.

The construction process of such hierarchical content representation starts from a low-level segmentation, such as color or texture based segmentation. Initially, the *object-region* tree consists only of root and elementary nodes. Elementary nodes are automatically constructed based on the low-level segmentation map with each node denoting a uniform region in the segmentation map. Composite nodes consist of groupings of elementary nodes or other composite nodes, hence the concept of hierarchical content description. Construction of higher level nodes with appropriate labeling usually requires either user interaction or learning from previous search patterns (learning by example). We describe a method for generation of composite nodes through learning by example in Section 2.3. Both elementary and composite nodes are associated with certain features and descriptors. For example, the mean color can be used for an elementary node while a composite node can be more accurately described by a color histogram when an object consists of parts with different colors.

2.2 Visual Content Matching and Similarity Measures

The proposed hierarchical content representation and labeling of composite nodes enables two types of visual queries: low level queries at the region level and high level queries at the semantic object level. High level queries are possible only if the representation of the images has composite (intermediate) nodes. Otherwise, only low-level queries will be allowed to match the query template to all neighboring combinations of elementary nodes in a given image. A procedure for formation and labeling of composite nodes is described in the next subsection. Here we discuss color and shape similarity measures.

2.2.1 Color Similarity Matching

Since elementary nodes generally correspond to regions with uniform color, we quantify color similarity between two elementary nodes by the distance between the mean colors of corresponding regions. However, to measure color similarity between two composite nodes or between a template and a combination of elementary nodes, we employ the histogram intersection method proposed by Swain and Ballard [23]. If T and O denote the color histograms of the query template and selected composite node or grouping of elementary nodes respectively, their histogram intersection is defined as [23]

$$H(O,T) = \frac{\sum_{j=1}^{n} \min(O_j, T_j)}{\sum_{j=1}^{n} T_j}$$

where we assume both histograms contain n bins. Note that $0< H(O,T) <1$, where $H(O,T)=0$ indicates no match. We say a match is established between the query template and combination of elementary nodes if the histogram intersection $H(O,T)$ is between a user specified color threshold t_c ($0<t_c<1$) and 1. While this similarity measure is fairly simple, it is remarkably effective in determining color similarity between images of multi-colored objects.

2.2.2 Shape Similarity Matching

We employ a region-based shape matching approach, where each region or combinations of neighboring regions (corresponding to elementary or composite nodes) constitute "potential" objects. The goal is to locate all translated, rotated and scaled instances of a template within a given image. The steps of our shape matching procedure can be summarized as follows [22]. First, the boundary of the query template, as well as each potential region or group of regions within the segmented image is approximated by B-splines. The joint points of the B-splines represent the boundary of the template and potential database object. A modal shape description [22, 24] is then employed to establish correspondences between the template boundary points and those of the potential object. These correspondences are used to compute affine parameters, which, in turn, are employed to map the image onto the template domain. Finally, the Hausdorff distance between the template contour and that of the mapped region or group of regions under investigation is computed to determine their similarity. The ultimate outcome is a table where each region or group of regions is given a similarity measure to the template under consideration. This table is normally arranged in increasing order from the most similar region or group of regions, as indicated by the smallest Hausdorff distance, to the least similar. Those regions or grouping of regions with a Hausdorff distance below certain given or user specified shape threshold t_s are pronounced as "similar" matches.

2.3 Object Extraction and Labeling through "Learning by Example"

In the context of this work, "learning" refers to storing frequently occurring combinations of regions, which correspond to meaningful objects, in the form of composite nodes with specific indexing and labeling information attached. We demonstrate the concept of "learning by example" by a specific example. Consider the situation where a user is interested in labeling all images that contain "red cars", and the user provides an example template, say a "red car," and the attributes to be used in similarity matching are "color" or "shape". Given that the hierarchical content tree descriptions of all images already exist, our approach then attempts to match the query template with the hierarchical content tree of all images using a top-down approach. If an image has not been labeled using this query template before, then, the search must consider all combinations of neighboring elementary nodes to determine if a match satisfying the user specified threshold (t_c or t_s, for color or shape) could be identified. If a match is established, then we form a composite node containing the matching combination of elementary nodes, information about the query template, and the similarity measure between them. This composite node, which is labeled as a match of "red car" template, would also contain, from this point on, the appropriate color and shape attributes of the "red car." This process constitutes a learning step that would not have taken place had we not performed the match. As a result, subsequent searches using the same query template and a similarity threshold t_c (for color) or t_s (for shape), which is tighter than the stored similarity measure, would immediately identify the composite node as a match using its shape and/or color attributes without the need to process all lower level nodes.

2.4 Computational Complexity of the First-Time Search Using Adjacency Matrix

The first-time low-level search for a given object attempts to match the query template to all valid combinations of elementary nodes in a given image. We now provide an analysis of the computational complexity of the first-time search, where valid combinations are defined as those combinations of nodes that correspond to neighboring regions in the image. Given an image that can be described by a complete graph [21] with N elementary nodes, corresponding to the elementary regions, and J links corresponding to neighboring relationships, the maximum number of possible links is $J=N(N-1)/2$. For a complete graph (worst case scenario), where all elementary nodes are interconnected, the maximum number of combinations that need to be examined is given by:

$$C = \sum_{k=1}^{N} \frac{N!}{k!(N-k)!} = \sum_{k=1}^{N} \binom{N}{k} = 2^N - 1$$

However, in general, the actual neighboring relations in images are local, and the "scene graph" can be modeled by a planar [21] instead of a complete graph. Therefore, the maximum number of links is $J = 3N-6$, instead of $N(N-1)/2$, yielding a much smaller set of valid combinations compared with the above worst case analysis. The algorithm given in Procedures 2.1 and 2.2 provides an iterative approach to generate valid combinations of neighboring elementary nodes in a given image from the adjacency matrix. Instead of testing all the combination of elementary nodes, this approach has greatly reduced the computational complexity of the first-time search of an object by avoiding testing invalid combinations.

Procedure 2.1: Generate all valid combinations that are subset of {1, 2…N}

For each node M from 1 to N

Generate valid sets that are subset of (M, M+1…N)

Procedure 2.2: Generate all valid combinations that are subsets of {M, M+1…N}

If M=N return {N}

For each node K from M+1 to N

If adjacent (M, K)==True

Make the adjacency matrix for nodes (K, K+1… N);

Call the procedure2.1 to generate all valid sets that are subsets of (K, K+1… N);

Generate valid combinations by combining M and each of the valid sets above;

In the above procedures, each number represents an elementary node in the initial content tree of the given image.

2.5 Advantages of the Hierarchical Content Tree

The hierarchical content tree provides a significant reduction in the number of combinations to be tested during subsequent searches after the initial labeling process is completed. For example, in Figure 1, any subsequent search for cars in the image, where a "car" object has been captured in a composite node, does not have to consider the combinations of elementary nodes that are "beneath" the composite car node, thereby significantly reducing the number of valid combinations to be tested [20].

Another advantage of the hierarchical content tree is that the composite nodes provide semantic object annotation (formation) without user intervention. For example, the composite node in Figure 1(e) captures the "car" object within the scene, which represents a level of semantic knowledge [20].

3 THRESHOLD-FREE MULTI-LEVEL SEGMENTATION

The object labeling through learning-by-example (see Section 2) assumes availability of a low-level image segmentation map to start from. However, the computation of a single segmentation map with an appropriate amount of granularity is a difficult problem as discussed below in Section 3.1. For this reason, the remainder of this section proposes a multi-level segmentation algorithm that is threshold-free. The integration of this multi-level segmentation with the learning-by-example framework (Section 2) is discussed in Section 4.

The proposed scheme generates a stack of images, which will be ordered in a uniformity tree. Each level of the uniformity tree is essentially a segmentation map of the image with a certain level of granularity. The top level of the uniformity tree is the coarsest. It consists of a single region that contains all the pixels of the whole image. On the other hand, the bottom level is the finest, where a single pixel could be a region.

3.1 Motivation for Multi-level Segmentation

In the learning by example framework (Section 2), some difficulties exist with the low-level image segmentation. Those are as follows:

- **Over-segmentation or under-segmentation**: Single level segmentation methods commonly result in either over-segmentation or under-segmentation because of lack of generic segmentation criteria and measures. The main problem with over-segmentation is the presence of too many small regions in the final segmentation map. Because every possible combination of neighboring regions has to be checked in the first-time search of an object, this will increase the computational complexity of the first-time search considerably. On the other hand, the difficulty with under-segmentation is that the accuracy of the segmented objects may not be satisfactory. Different objects may be merged with each other or with portions of the background resulting in inaccurate labeling.
- **Objects are in different scale:** Objects in images can be in very different scales. Hence image segmentation procedures, using single size criteria to remove small regions, could lead to smaller objects with small region size being accidentally merged to larger neighbors, e.g. “eaten” by big neighboring objects with large region size.

Given these problems, we propose a threshold-free multi-level image segmentation approach for low-level image analysis to yield “uniformity trees” as the basis for the learning by example framework. Instead of generating a single segmentation

map for a given image, multiple segmentation maps are generated for any color image forming the uniformity tree that will be used in similarity searching.

3.2 Multi-level Segmentation Algorithm

The following procedures describe the bottom-up process of the multi-level segmentation process. The main idea is to merge the most similar regions in a lower level segmentation map resulting in a higher level (coarser) segmentation map. The algorithm starts with the original image that forms the bottom level (finest) and repeatedly merges the most similar regions to form the intermediate levels until it reaches a single region at the top level. The measuring of the similarity depends on the feature selected. In this work, color is adopted as the measuring feature.

Initialization:

Find the initial homogeneous regions

For each pair of neighboring regions A and B

Compute the distance d(A,B) between A and B according to the selected similarity criteria

Insert d(A,B) into the distance list

Merging:

Repeat

Find the region pair with the minimum distance

Merge the regions A and B

Update the distance list

Until a single region is obtained

By using the above algorithm, it is possible to generate a stack of images $I_M,\ldots,I_1$, starting from the initial homogeneous regions and iteratively combining the regions in the image I_{j-1}, from the regions of image I_j. Let $\{R_{ij}\}$, $i=1,\ldots,N_j$, be a set of regions at level j, the image at level j can be represented as a mutually exclusive and collectively exhaustive set of regions as

$$I_j = \bigcup_{i=0}^{N_j} R_{ij} \qquad j = 1,\cdots,M$$

Note that a homogeneous region may consist of a single pixel if the colors of all its neighboring pixels are not "most similar" in the palette. Also, note that for any m in $\{1,...,N_j\}$, an n exits in $\{1, ... ,N_{j-1}\}$ so that $R_{mj} \subset R_{nj-1}$ is true for any j in $\{2,...,M\}$.

The distance list contains the color distance between any two regions in the segmentation map at the current level. The step of updating the distance list is performed by removing the values corresponding to the distances between colors of the regions that were merged, as well as adding the new values of color distances between the merged region and its neighbors in the new level segmentation map. After each merging step, the size of the distance list decreases by at least one since at least one color distance between two merged regions is removed. It is possible to have multiple region pairs that possess the same color distance. In our current algorithm, we choose one of those pairs randomly to merge. Although all of these region pairs can be merged in a single step, we prefer to merge one pair a step since it simplifies the merging process.

3.3 Uniformity Tree

A uniformity tree consists of all the segmentation maps for a given image generated by the multi-level segmentation algorithm. By stacking all the segmentation maps in the reverse order of which they are generated, each level of the uniformity tree provides a representation of the given image with different granularity. As seen in Figure 2, the lowest, i.e. the finest, level of the tree could be the original image itself, where each pixel is a single region, and the highest, i.e. the coarsest, level of the tree, is a segmentation map with only one region containing all the pixels in the image. As such, the uniformity tree does not only eliminate the need for an explicit threshold for the segmentation, but also provides a "coarse" to "fine" representation of the given image. An example of multi-level segmentation is shown in Figure 3.

3.4 Merging Process

Given a segmentation map in the uniformity tree, two neighboring regions A and B in 4-neighborhood are merged to produce a new region R in the higher level of the uniformity tree, if they are the most similar regions in the lower level according to the selected similarity feature. Let $d(A,B)$ be the similarity distance between region A and B based on the selected similarity feature, mathematically this process is equivalent to:

If $d(A, B) < d(M, N)$ *for all* (M, N)

Then $R = A \cup B$

where *M, N* are the two different regions in the given segmentation map and *R* is a single region in the segmentation map at the level above.

3.4.1 Similarity Feature

In this work, we use color as the similarity feature and propose five different methods in the next subsection for merging the regions. In all these methods, *L*a*b* color space is adopted for its inherited merits in overcoming undesirable effects caused by varied lighting conditions and achieving more robust an illumination invariant segmentation. To this effect, we compute the Euclidean distance in the *L*a*b* color space as the similarity distance between two regions. Given two colors (*k, l*) in the color palette of the image, the distance is computed as

$$E(k,l) = \sqrt{(k_L - l_L)^2 + (k_a - l_a)^2 + (k_b - l_b)^2}$$

The simplest method is to assign a color to each region and then find the distance between the regions using the above Euclidean color distance, i.e.

$$d(A,B) = E(C_A, C_B)$$

where C_A and C_B represent the colors of region A and B respectively. Two neighboring regions *A* and *B* form a new region *R*, if they have the closest colors with respect to the Euclidean distance in *L*a*b* color space. In other words, the two neighboring regions are merged into a new region if the Euclidean distance between their colors has the smallest possible value in the corresponding segmentation level. This scheme is called as "the closest colors in the same neighborhood".

Instead of using a single color as the similarity feature, as proposed by Deng et al in [18], each region can also be represented by its color codebook that keeps all the colors and the frequencies of the color in the region. Given an image region *A*, the codebook can be described as

$$\{(C_a, P_a) \mid 1 \le a \le N_a\}$$

where N_a corresponds to the number of colors in the codebook for the region, C_a is the a^{th} color in the codebook of region *A* and P_a is the occurrence frequency of C_a.

3.4.2 Region Merging Methods

In this work, five different methods using the two above similarity features have been studied for the step of region merging. In the first two methods, regions are represented by single colors and "the closest colors in the same neighborhood" scheme is used to merge the regions. The two differ only in the way of assigning color for the new region. Method 3 uses color codebooks as the similarity feature and regions with the most similar color codebook are merged. In the remaining two methods, the sizes of the regions are weighted in using their color features to determine regions to be merged, where single color and color codebooks are adopted as the color feature in method 4 and 5 respectively.

Method 1: Dominant Color

"The closest color in the same neighborhood" scheme is used. In particular, the dominant color is computed for each region and the similarity distances are calculated based on the dominant colors. Regions with the most similar dominant colors in the given level are merged into a new region.

$$d(A,B) = E(C_A, C_B)$$

where C_A and C_B represent the dominant color of region A and B respectively. The color of the new region is found by computing the dominant color in the new region.

Method 2: Color of the larger region

"The closest color in the same neighborhood" scheme is used. Initial regions have single homogeneous colors. After merging step, the color of the larger region is assigned as the color of the new region.

Method 3: Closest Color Codebooks

The distance between two neighboring regions A and B is computed using the color codebooks as:

$$d(A,B) = \sum_{1 \le a \le Na} D[(C_a, P_a),B] + \sum_{1 \le b \le Nb} D[(C_b, P_b),A]$$

where

$$D[(C_a, P_a),B] = |P_a - P_b| \cdot E(I_a, I_k)$$

and k is the closest color to C_a in B and computed by

$$k = \text{argmin}_{1 \le b \le Nb} E(C_a, C_b)$$

For each color in A, its closest color in B is found and their distance is calculated as $D[(C_a, P_a),B]$, where the color frequencies are used as weighting factors.

Method 4: Color and Size

Given region A and B, where the size of A is smaller than the size of B, the distance is computed as:

$$d(A,B) = E(C_a, C_b). S(A)$$

where $E(C_a, C_b)$ is the Euclidean distance between color C_a and C_b and $S(A)$ is the size of the smaller region.

Method 5: Color Codebook and Size

Given two regions A and B, where the size of A is smaller than the size of B, each color in A is replaced by its most similar color k in B, which is found by

$$k = argmin_{1 \le b \le Nb} E(C_a, C_b)$$

Then, the similarity distance between the two regions is computed as:

$$d(A,B) = \sum_{1 \le a \le Na} E(C_a, C_k), P_a$$

The codebook of the merged region is computed by updating the frequency of the color k in the codebook of B with ($P_k + P_a$).

Figure 7 provides a comparison of the above methods by showing the errors produced in the merging step. Error is computed as the sum of differences between the original and assigned color after the merging of all individual pixels. As seen from the figure, methods 1 and 3 have very similar errors, i.e. assigning dominant color and assigning a new color codebook gives similar results. That is also the case for method 4 and method 5 except that larger regions have dominancy. Therefore, assigning a single color can be adopted for efficiency purposes. In our work, Method 5 is selected for merging the regions, since larger objects are more sensitive to human eyes and deserve some dominancy in the merging process.

4 OPTIMIZED SEARCH WITH MULTI-LEVEL SEGMENTATION

This section integrates multi-level segmentation with the hierarchical content tree representation to improve both the speed and accuracy of object labeling. The first-time search of an object will now be performed using the uniformity tree from the coarsest to the finest level. The coarser levels of the uniformity tree contain under–segmented layers, whereas the finer levels contain over-segmented layers. In this way larger objects can be found at the coarse levels with high speed, and smaller objects can be found at the fine levels with high accuracy as desired.

4.1 Optimized Search Using the Uniformity Tree

As described in the previous section, the construction of the uniformity tree is a bottom-up process, where a coarser level of the segmentation map is constructed from the finer level by merging neighboring regions with the most similar colors. However, when we search for an object, we perform the matching, using the uniformity tree, in a top-down fashion. We attempt to find the given object in the coarsest level possible to avoid testing unnecessary combinations at the finer levels.

The initial search of an object is performed by testing every valid combination of the elementary nodes, which correspond to uniform regions in a segmentation map in each level of the uniformity tree. From the construction process, we know that there are redundancies in region combinations at different levels. The redundancies lead to unnecessary tests of regions at the finer level that have already been tested as different combinations at coarser levels. It is obviously helpful to reduce the computation complexity by avoiding these redundant tests in different levels of uniformity tree. With this in mind, we propose Procedure 4.1 as a searching strategy to avoid all the unnecessary tests in a small price of sets operations.

Procedure 4.1: Search an object against a uniformity tree

For level M= 1 to N

Call procedure 2.1 to generate the set of all valid combinations $\boldsymbol{S_M}$ *for segmentation map in level M;*

In $\boldsymbol{S_M}$*, find the set of all the combinations* T_M*, each combination in which has an equivalent combination in* $\boldsymbol{S_{M-1}}$*;*

Perform matching test against each combination in $\boldsymbol{S_M}$*-* T_M*;*

4.2 Computational Complexity of Search Using Multi-level Segmentation

The complexity of the multi-level segmentation algorithm depends on the initial number of homogeneous regions. The size of the initial distance list (N) can be *4*(image size)* in the worst case since we use 4-neighborhood. For each merging step, we have at most N iterations to find the minimum value and N iterations to update the distance list. Hence, at each merging step, the size of the list decreases by at least one. So, the complexity of the overall algorithm is $O(N^2)$. However, since the operations in the inner most iteration are relatively simple, the multi-level segmentation algorithm has a much lower complexity when compared to the sophisticated single level segmentation algorithm previously adopted for low level image analysis [20].

We now analyze the computational complexity of searching the uniformity tree using the proposed procedure. First, let's look at an illustrative example: assume a uniformity tree only has two levels, with region R_{11} and R_{21} in the coarse

level and R_{12}, R_{22}, R_{32}, R_{42} in the fine level respectively, where $R_{11} = R_{12} \cup R_{22}$ and $R_{21} = R_{32} \cup R_{42}$. For simplicity, it is also assumed that all the regions are inter-connected, therefore, the valid region combinations at the coarse and fine levels are:

Coarse Level:

$\{R_{11}\}$, $\{R_{21}\}$, $\{R_{11}, R_{21}\}$

Fine Level:

$\{R_{12}\}$, $\{R_{22}\}$, $\{R_{32}\}$, $\{R_{42}\}$,

$\{R_{12}, R_{22}\}$, $\{R_{12}, R_{32}\}$, $\{R_{12}, R_{42}\}$, $\{R_{22}, R_{32}\}$, $\{R_{22}, R_{42}\}$, $\{R_{32}, R_{42}\}$,

$\{R_{12}, R_{22}, R_{32}\}$, $\{R_{12}, R_{22}, R_{42}\}$, $\{R_{12}, R_{32}, R_{42}\}$, $\{R_{22}, R_{32}, R_{42}\}$;

$\{R_{12}, R_{22}, R_{32}, R_{42}\}$

Note that $\{R_{11}\}$, $\{R_{21}\}$ and $\{R_{11}, R_{21}\}$ are equivalent to $\{R_{12}, R_{22}\}$, $\{R_{32}, R_{42}\}$ and $\{R_{12}, R_{22}, R_{32}, R_{42}\}$ respectively. By using the strategy in Procedure 4.1, the combinations $\{R_{12}, R_{22}\}$, $\{R_{32}, R_{42}\}$ and $\{R_{12}, R_{22}, R_{32}, R_{42}\}$, will be skipped since they have already been tested at the coarse level. So the resulting combinations that will be tested in this example are:

$\{R_{11}\}$, $\{R_{21}\}$, $\{R_{11}, R_{21}\}$,

$\{R_{12}\}$, $\{R_{22}\}$, $\{R_{32}\}$, $\{R_{42}\}$,

$\{R_{12}, R_{32}\}$, $\{R_{12}, R_{42}\}$, $\{R_{22}, R_{32}\}$, $\{R_{22}, R_{42}\}$

$\{R_{12}, R_{22}, R_{32}\}$, $\{R_{12}, R_{22}, R_{42}\}$, $\{R_{12}, R_{32}, R_{42}\}$, $\{R_{22}, R_{32}, R_{42}\}$,

which are exactly the same as the complete combinations in the fine level.

For a more common case with N levels of segmentation map in the uniformity tree, it is easy to prove that our searching procedure will perform exactly the same number of tests as if we test only the combinations in the finest level segmentation map. Assume in the worst case, we have N levels in the uniformity tree, and the number of regions in each level decrease by one. From the discussion in Section 2.4, in the worst case, the number of total combinations to search a single segmentation map in level m would be:

$$C_m = 2^m - 1$$

Then the total number of combinations to be tested in searching the whole uniformity tree would be:

$$C = C_1 + \sum_{m=2}^{N} (C_m - C_{m-1}) = (2^1 - 1) + \sum_{m=2}^{N} (2^m - 1) - (2^{m-1} - 1) = 2^N - 1$$

From the analysis above, the advantage of our searching strategy becomes more obvious, e.g. in the worst case when we have to search all levels of the uniformity tree, it is still equivalent to searching a single segmentation map at the finest

level. However, when a match is found at level *N-k* prior to reaching the finest level *N* of the uniformity tree, the total number of combinations to be tested will be:

$$C' = C_1 + \sum_{m=2}^{N-k}(C_m - C_{m-1}) = 2^{N-k} - 1$$

which is much less than 2^N*-1*, the total number of combinations in the finest level segmentation map.

5 EXPERIMENTAL RESULTS

Figures 3, 4, 5, and 6 demonstrate the results of our proposed multi-level segmentation and object based image labeling algorithms on real life images. All original color images are of RGB type, where each channel is quantized to 8 bits/pixel. In the multi-level segmentation experiments, all original images are converted to *L*a*b** color space to initiate the construction of the corresponding uniformity trees. The attributes used for matching in the image labeling experiments are "color" and "shape" using the corresponding similarity measures discussed section 3. The segmentation maps are shown in color for reader's convenience.

5.1 Multi-Level Segmentation

Figure 3 demonstrates the multi-level segmentation process on a "car" image, where we have focused specifically on the construction process of the uniformity tree. The original image is shown in Figure 3a. It possesses 10,000 regions and has been marked by a level label of 10,000. Figures 3b through 3i show intermediate stages of the uniformity tree with their level and region count indicated directly beneath them. It should be noted that the level label decreases as we progress from 3a to 3j indicating that the segmentation possesses less and less regions. Finally, Figure 3j shows the last phases of the construction of the uniformity tree, with the image containing only two regions. Notice in this level, the "car" object has been completely separated from the background, thereby simplifying the shape matching process significantly as can be seen later in Section 5.2. Similar results are observed in Figure 4, where the multi-level segmentation process has been applied to "mountain" and "portrait" scenes respectively.

Besides demonstrating the construction of the uniformity tree, Figures 3 and 4 outline the benefits of the multi-level segmentation in reducing the computational complexity of the initial search for a given object. For instance, consider the situation where the user is searching for "cars" in images using the "shape" attribute. By traversing the uniformity tree and beginning at the least regions level, our proposed approach will test far fewer combinations to extract the car than an

approach with a single segmentation map that contains a certain number of regions. In the case of Figure 3j, the match will be obtained at the very first level, where the shape similarity matching approach will test only 1 combination.

5.2 Object Extraction and Labeling Experiments

Figures 5 and 6 demonstrate our proposed techniques on several images using the "shape" and "color" attributes, respectively. Figure 5 is made up of four car images suitable for shape matching queries, while Figure 6 contains four portrait images selected for color matching queries. The query example template is shown at the top of each figure. In each Figure, we show (a) the original image, (b) one level of the segmentation map in the uniformity tree, (c) the query by example results obtained by searching the original image, and (d) the resulting hierarchical content tree with the composite node labeled from the learning process. The segmentation level shown in (b) is the one where a match has been established as the search traverses the uniformity tree to finer and finer resolutions.

While the hierarchical content description scheme and the learning concept provide the basis for automatic content analysis for object based image labeling, Figures 5 and 6 clearly demonstrate the benefit of applying the multi-level segmentation. By providing "coarse" to "fine" segmentation levels that can be searched in a top-down fashion, the multi-level segmentation significantly reduces the number of tested combinations in labeling for a given object within a scene.

5.3 Performance Evaluation

Table 1 provides a summary of the performance evaluation of our proposed algorithms based on the object search and formation tests described in Section 5.2. In the table, we show two groups of results. The first set documents our search results using only a single segmentation level. The number of regions found within the segmentation map, and the number of potential first search for labeling and subsequent search combinations after the object has been labeled are also shown for comparison. Note that the number of combinations tested for a successful match is significantly reduced between the initial search and any subsequent search for all images due to the formation of the composite node in the hierarchical content tree.

On the other hand, the second set captures the search results obtained by utilizing our proposed top-down search strategy against the uniformity tree obtained from the multi-level segmentation. For each image, we show the segmentation level where a match has been established and the number of regions found within, the actual number of combination tested in the initial and subsequent searches for the "car" and "face" objects. Note that the number of combinations tested between initial and subsequent searches is reduced in a similar fashion to the single level segmentation due to the hierarchical content

representation. More importantly, the number of initial combinations tested is significantly reduced when comparing the single and multi-level segmentation results due to the presence of the multi-level segmentation maps.

5.4 Image Labeling Experiment on a Larger Dataset

To further test the performance of the proposed methods for object based image labeling, experiments were performed on a larger dataset containing about 200 color images using multiple object templates. Rather than using pictures taken by researchers in a special setup, most of the images were collected from the internet or other sources and were taken by unknown authors. Figure 8(a) provides a subset of the image database (20 images) numbered from one to twenty in a raster scan fashion starting from left to right and top to bottom. Images 1-10 and 11-15 represent "Sedans" and "SUVs" respectively. Image 16 corresponds to a scene with multiple cars. Images 10, 17-19 show various occlusions and finally Image 20 provides a significant rotation out of the image plane. On the other hand, Figure 8(b) shows the two templates used in these experiments and the results of the image labeling using these two templates.

Table 2 clearly demonstrates the computational advantages of our proposed multi-level segmentation when compared with its single level predecessor, where the number of combinations tested to arrive at the correct labeling was reduced significantly. This reduction was achieved as a result of the ability of our algorithm to perform the labeling in a top-down fashion proceeding from the "coarsest" segmentation map (least amount of regions) to the "finest" one (most amount of regions) and stopping along the path once a match is established.

Table 3 provides a quantitative evaluation of our proposed algorithm using the images displayed in Figure 8(a). In the table, we show the image index, type, brief description and its corresponding Hausdorff distance using the "Sedan" and "SUV" templates respectively. Note that Image 16 contains two cars hence its results are displayed in two separate rows. As seen from Table 3, the top three matches for the "Sedan" template (Images 1, 8, and 2) were "Sedans". In addition among the 12 "Sedans" (Images 1-10 and 16) displayed in Figure 8(a), Images 7 and 9 were the most dissimilar due to their rotation out of the image plane. These rotations caused them to become less similar than Images 13 and 15 (SUV images). Similar results were observed for the "SUV" template for the top 3 matches, where the least dissimilar "SUVs" (Images 11 & 14) displayed rotations out/in the image plane respectively. However, since a typical user would be interested in similar matches arranged in descending order, we believe this approach provides a suitable solution for achieving the task.

In addition to the quantitative discussion above, Table 3 indirectly displays potential challenges and failure modes. In essence, our algorithm failed to recognize/label the occluded Sedans (see Images 17-19), the significantly occluded Sedan

(see Image 10), and the Sedan significantly rotated out of the image plane (see Image 20). Furthermore, Images 7, 9, 11 and 14, while labeled correctly, displayed degraded similarity performance due to their slight rotations in/out of the image plane. These potential shortcomings are outcomes of the 2D shape matching algorithm (see [11]). While, it is capable of handling translations, rotations within the image plane, and isometric scaling, it is not well suited for general affine transformations especially those with significant shearing and perspective projections.

6 CONCLUSION

The proposed object based image labeling and description provides a basis for applications such as object based image searching, browsing and retrieving as well as object based image indexing for effective management of large image collections. The multi-level segmentation (uniformity-tree) based approach has several advantages over that based on a single level segmentation. These are: 1) Over-segmentation or under-segmentation either increases the computation complexity of image labeling or results in inaccurate labeling results when small objects in images are hidden by the under-segmentation; 2) Simple thresholding results in objects with small region size "eaten" by neighboring objects with larger regions size; 3) Complexity of the first search of any object in an image is expensive although the efficiency of subsequent search is greatly improved by the learning behavior. These are demonstrated by our experimental results.

Potential extensions of this work may be: 1) to include perspective projections where objects rotated in/out of the image plane (see Image 20), 2) to provide a solution for partially occluded objects similar (see Image 17-19), and 3) to extend to digital video object labeling for video content analysis, object tracking and motion analysis.

ACKNOWLEDGEMENTS: This material is based upon work supported by the National Science Foundation under Grants IIS-9820721 and INT-9731007 to the University of Rochester.

REFERENCES

1. A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content- Based Image Retrievals at the End of the Early Years,” IEEE Transactions on Patt. Anal. and Mach. Intell., 22(12)(2000) 1349-1380.
2. Y. Rui, T. S. Huang, and S.-F. Chang, “Image retrieval: current techniques, promising directions and open issues”, Journal of Visual Communication and Image Representation, 10(4)(1999) 39-62.
3. F. Idris and S. Panchanathan, “Review of image and video indexing techniques”, Journal of Visual Communication and Image Representation, 8(2)(1997) 146-166.
4. M. De Marsicoi, L. Cinque, and S. Levialdi, “Indexing pictorial documents by their content: a survey of current techniques”, Image and Vision Computing, 15(2)(1997) 119-141.
5. A. Jain and A. Vailaya, “Shape based retrieval: a case study with trademark image databases”, Pattern Recognition, 31(9)(1998) 1369 – 1390.
6. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, “Efficient and effective querying by image content”, Journal of Intelligent Information Systems, 3(3-4)(1994) 231-262.
7. A. Pentland, R. W. Picard, and S. Sclaroff, “Photobook: content-based manipulation of image databases”, International Journal on Computer Vision, 18(3)(1996) 233-254.
8. J. R. Smith and S.-F. Chang, "VisualSEEK: A fully automated content-based image query system", Proceedings of ACM Multimedia 96, 1996.
9. J. R. Smith and S.-F. Chang, "Visually searching the web for the content", IEEE Multimedia Magazine, 4(3)(1997) 12-20.
10. V. E. Ogle and M. Stonebraker, “Chabot: retrieval from a relational database of images”, Computer, 28(9)(1995) 40-48.
11. C. Okon, “Image recognition meets content management for authoring, editing and more”, Advanced imaging, 10(7)(1995).
12. T. P. Minka and R. W. Picard, “Interactive learning using a society of models”, Pattern Recognition, 30(4)(1997) 565-581.
13. Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool in interactive content-based image retrieval", IEEE Trans. On Circuit. Sys. Video Tech. 8(5)(1998) 644-655
14. J. Dowe, "Content-based retrieval in multimedia imaging", Proceedings SPIE Storage and Retrieval for Image and Video Database, 1993.
15. N. R. Pal and S. K. Pal, “A Review on Image Segmentation Techniques”, Pattern Recognition, 26(9)(1993) 1277-1294.
16. T. Gevers and F. C. A. Groen, “Segmentation of color images”, Proceedings of 7th Scandinavian Conference on Image Analysis, 1991.
17. C.-S. Fuh, S.-W. Cho and K. Essig, “Hierarchical Color Image Region Segmentation for Content-Based Image Retrieval System”, IEEE Transactions on Image Processing, 9(1)(2000) 156-162.
18. Y. Deng, B. S. Manjunath, H. Shin, “Color Image Segmentation”, Proc. Of IEEE Conf. On Computer Vision and Pattern recognition (CVPR99), 1999.

19. E. Saber, A. M. Tekalp, and G. Bozdagi, “Fusion of Color and Edge Information for Improved Segmentation and Edge Linking”, Image and Vision Computing, 15(1997) 769-780.
20. Y. Xu, E. Saber, and A. M. Tekalp, “Hierarchical Content Description and Object Formation by Learning”, Proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL'99) 1999.
21. M. Swamy and K. Thulasiraman, “Graphs, Networks, and Algorithms”, Wiley, NY, 1981.
22. E. Saber and A. M. Tekalp, “Region-based shape matching for automatic image annotation and query by example”, Journal of Visual. Comm. & Image Rep., 8(1)(1997) 3-20.
23. M. Swain and D. Ballard, “Color indexing”, International. Journal of. Computer. Vision, 7(1)(1991) 11-32.
24. L. Shapiro and J. Brady, “Feature-based correspondence: an eigenvector approach”, Image and Vision Computing, 10(1992) 283-288.
25. S. Santini, A. Gupta, and R. Jain, “Emergent semantics through interaction in image databases”, IEEE Transactions on Knowledge and Data Engineering, 13(3)(2001) 337-351.
26. A. Vailaya, A. Jain, and H.J. Zhang, “On image classification: city images vs. landscapes”, Pattern Recognition, 31(12)(1998) 1921-1935.
27. C. Colombo, A. Del Bimbo, P. Pala, “Semantics in visual information retrieval”, IEEE Multimedia, 6(3)(1999) 38-53.
28. T. Gevers and A.W.M. Smeulders, “PicToSeek: Combing Color and Shape Invariant Features for Image Retrieval,” IEEE Transactions on Image Processing, 9(1)(2000) 102-119
29. R. Zhao and W.L. Grosky, “From Features to Semantics: Some Preliminary Results,” 2000 International Conference on Multimedia and Expo, 2000
30. A. Del Bimbo, “Semantics-Based Retrieval by Content,” 2000 International Conference on Image Processing, 2000

VITAE

Yaowu Xu is currently a senior engineer at On2 Technologies Inc. He received the BS degree in Physics in 1993, the MS and PhD degrees in Nuclear Engineering in 1998 from Tsinghua University, Beijing, China. He also holds the MS degree in Electrical and Computer Engineering from University of Rochester. NY. His research interest includes image segmentation and annotation, content-based image analysis and retrieval, pattern recognition and digital video compression.

Pinar Duygulu is a PhD student in Computer Engineering Department at the Middle East Technical University. She received her BSc and MSc in computer engineering from Middle East Technical University, Turkey. She was a visiting scholar at UC Berkeley in Computer Science Division during February 2001-May 2002. She received the best paper in cognitive vision award in ECCV 2002. Her research interests include computer vision, document analysis and browsing and retrieval in image collections. She is a member of the IEEE Computer Society and the Turkish Pattern Recognition and Image Analysis Society.

Eli Saber joined Xerox in 1988 and is currently a Product Development Scientist and Manager in the Print Engine Development Unit at Xerox Corporation. He received the BS degree in Electrical and Computer Engineering from the University of Buffalo in 1988, and the MS and PhD degrees in the same discipline from the University of Rochester in 1992 and 1996 respectively. He is currently an adjunct faculty member at the Electrical & Computer Engineering Departments of the University of Rochester and the Rochester Institute of Technology responsible for teaching graduate coursework in signal, image and video processing and performing research in digital libraries and watermarking. His research interests include color image processing, image/video segmentation, content-based image/video analysis and retrieval, computer vision, and watermarking.

He was the recipient of the Gibran Khalil Gibran Scholarship and of several prizes and awards for outstanding academic achievements from 1984 to 1988, as well as the Quality Recognition award in 1990 from The Document Company, Xerox. He is a member of the Electrical Engineering Honor Society, Eta Kappa Nu, the Imaging Science and Technology Society, and a senior member of the Institute of Electrical and Electronic Engineers. He holds a number of conference and journal publications in the field of signal, image, and video processing.

A. Murat Tekalp received the M.S. and Ph.D. degrees in electrical, computer, and systems engineering from Rensselaer Polytechnic Institute (RPI), Troy, New York, in 1982 and 1984, respectively. From December 1984 to August 1987, he was with Eastman Kodak Company, Rochester, New York. He joined the Electrical and Computer Engineering Department at the University of Rochester, Rochester, New York, in 1987, where he is currently a Distinguished Professor. He is also with the Koc University, Istanbul, Turkey since June 2001. His current research interests are in digital image and video processing, including image/video restoration, video segmentation, object tracking, video coding, content-based video description, and secure media.

Prof. Tekalp received the NSF Research Initiation Award in 1988, and named as Distinguished Lecturer by IEEE Signal Processing Society in 1998. He has chaired the IEEE Signal Processing Society Technical Committee on Image and Multidimensional Signal Processing (Jan. 1996 - Dec. 1997) and is a founding member of the Technical Committee on Multimedia Signal Processing. He has served as an Associate Editor for the IEEE Trans. on Signal Processing (1990-1992) and IEEE Trans. on Image Processing (1994-1996). He was appointed as the Special Sessions Chair for the 1995 IEEE International Conference on Image Processing and Technical Program Co-Chair for IEEE ICASSP 2000 in Istanbul, Turkey. He is the founder and first Chairman of the Rochester Chapter of the IEEE Signal Processing Society. He was elected as the Chair of the Rochester Section of IEEE in 1994-1995.

At present, he is the Editor-in-Chief of the EURASIP journal on Image Communication published by Elsevier Science. He is also the General Chair of IEEE International Conf. on Image Processing, Rochester, NY in September 2002. He authored the Prentice Hall book Digital Video Processing (1995). Dr. Tekalp holds five US patents. His group contributed technology to the ISO/IEC MPEG-4 and MPEG-7 standards.

Fatos T. Yarman-Vural received the BSc degree with honors in electrical engineering from the Technical University of ISTANBUL in 1973, the MSc degree in electrical engineering from Bogazici University in 1975 and the PhD degree in electrical engineering and computer science from Princeton University in 1981. From 1981 to 1983, she was research scientist in Marmara Research Institute in Turkey. From 1983 to 1985, she was a visiting professor at Drexel University. From 1985 to 1992, she was a technical and deputy manager at Yapitel Inc. Since 1992, she has been a professor in Computer Engineering Department of Middle East Technical University (METU). Her research area covers computer vision, image processing, pattern recognition and artificial intelligence. She has been involved in teaching, consulting and organizing conferences in these areas.

Dr. Yarman-Vural was the chair woman in the Computer Engineering Department between 1996 and 2000. Currently, she is assistant to the President at METU. She is a senior member of IEEE and member of Turkish Informatics Foundation, Turkish Information Association and chamber of Electrical Engineers in Turkey.

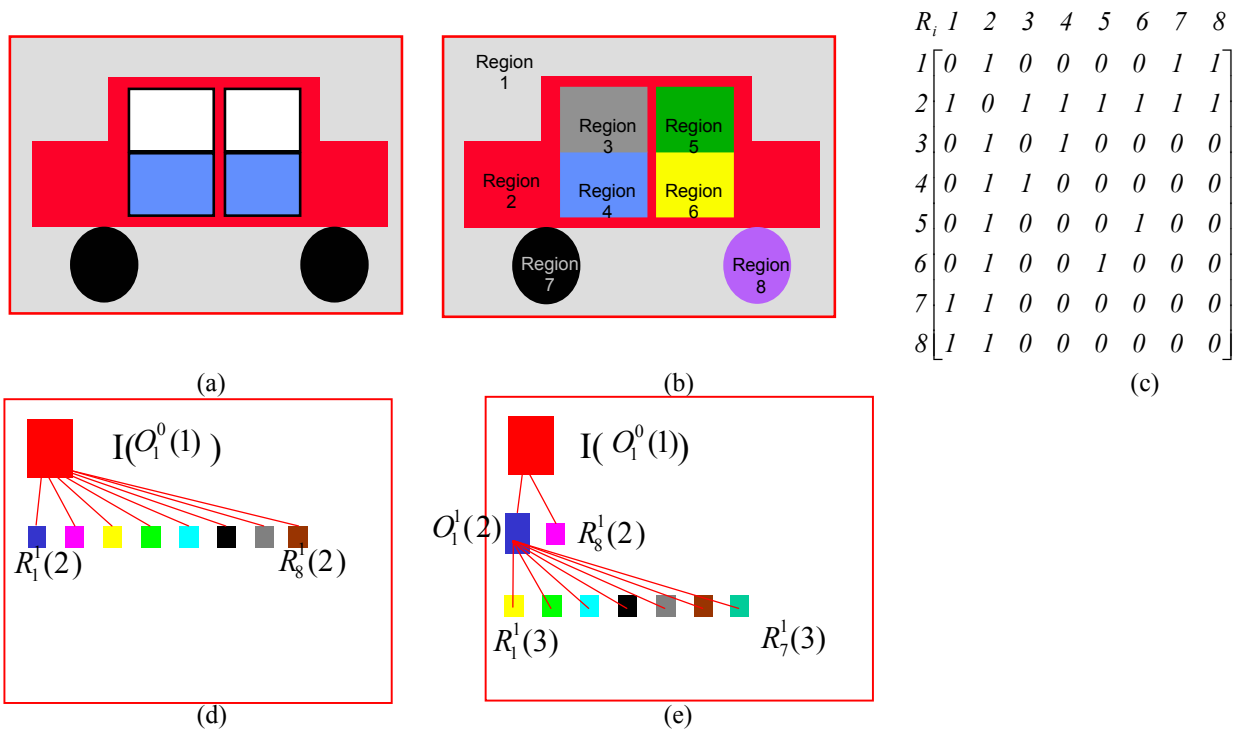


Figure 1: Hierarchical object description. a) Original image; b) Final low-level segmentation; c) Adjacency matrix; (d) Parent-child relationship tree with only elementary nodes, where the big square represents the whole image and small squares represent elementary nodes; and (e) Hierarchical content tree with an intermediate semantic-layer (composite nodes), where rectangle represents the composite node.

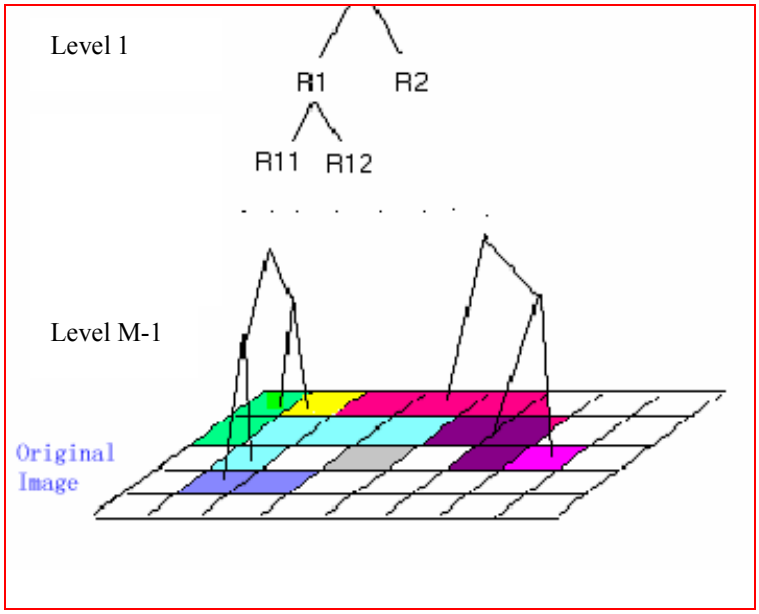


Figure 2: Uniformity Tree

(a) Original Image

Number of regions = 9999 level = 9999

(b) Level = 5000

Number of regions =5000

(c) Level = 500

Number of regions = 500

(d) Level = 50

Number of regions = 50

(e) Level = 10

Number of regions = 10

(f) Level = 6

Number of regions = 6

(g) Level = 5

Number of regions = 5

(h) Level = 4

Number of regions = 4

(i) Level = 3

Number of regions = 3

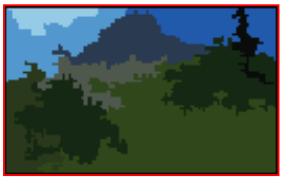
(j) Level = 2

Number of regions = 2

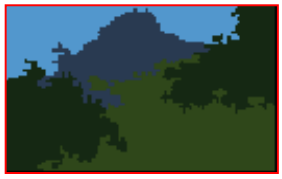
Figure 3: Results of a multi-level segmentation example.

(a1) Original Image

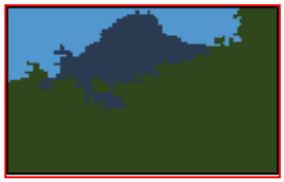
(b1) Level=16



(c1) Level=10

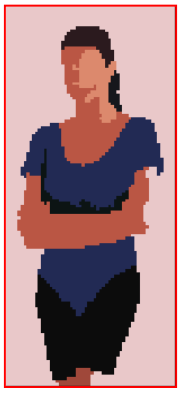


(d1) Level =5

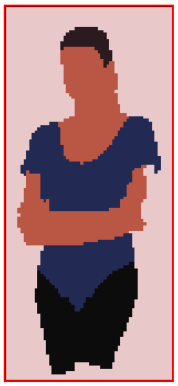


(e1) Level=3

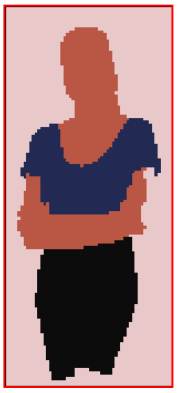
(a2) Original Image



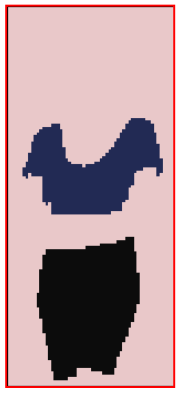
(b2) Level=11



(c2) Level=7



(d2) Level =5



(e2) Level=3

Figure 4: More results of multi-level segmentation tests.

Query template

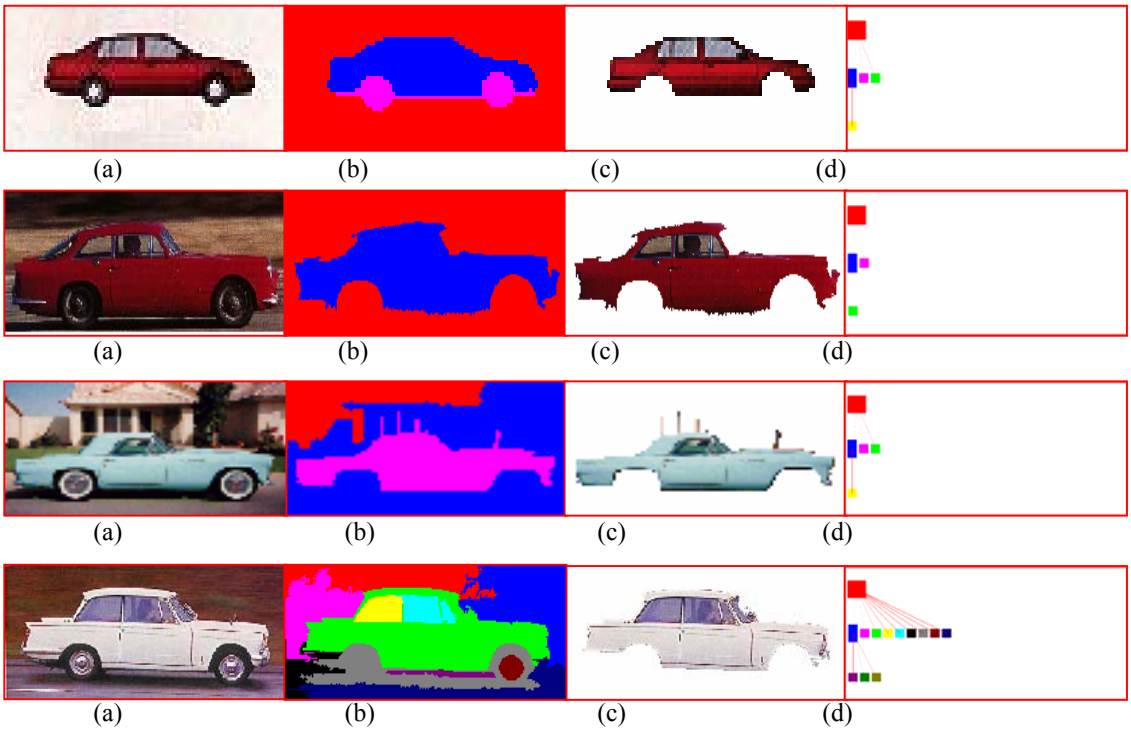


Figure 5: Results of image labeling tests on car images using shape matching: a) Original image; b) Final low-level segmentation using multi-level segmentation scheme; c) Semantic segmentation matching the query template; d) Hierarchical content tree with a composite node representing the segmented car.

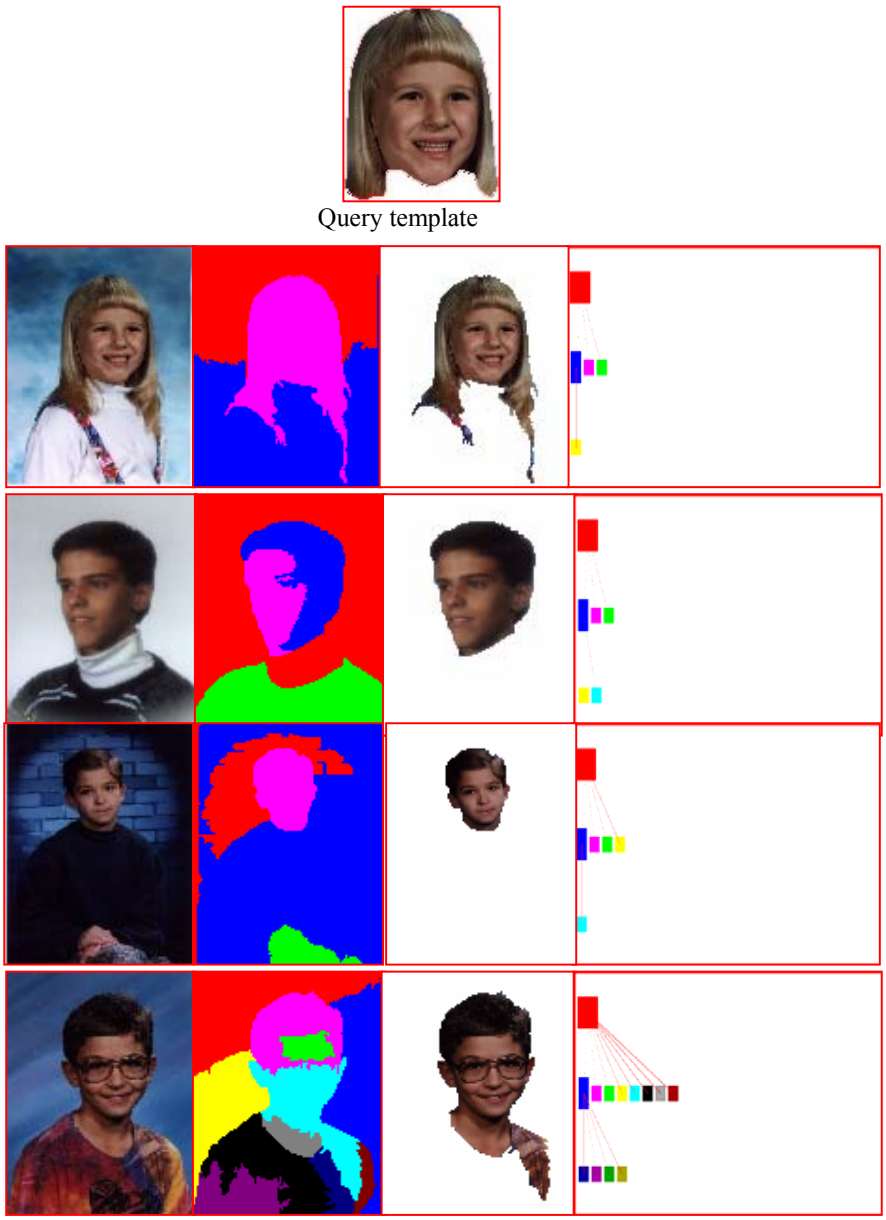
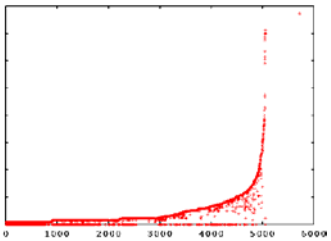
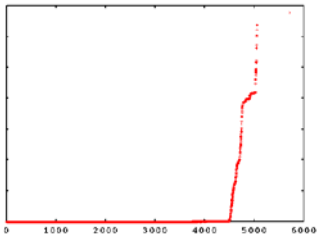


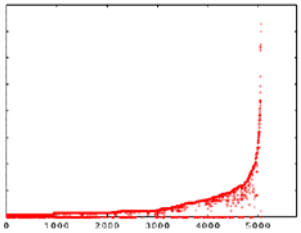
Figure 6: Results of image labeling tests on face images using color matching: a) Original image; b) One level segmentation map in uniformity tree; c) Semantic segmentation matching the query template; d) Hierarchical content tree with a composite node representing the segmented face.



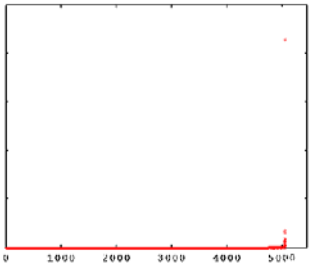
Method1



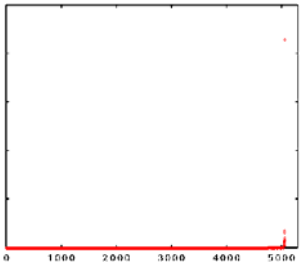
Method2



Method3



Method 4



Method5

Figure 7: Comparison of the methods proposed for merging regions for the image in Figure3.

(a) Subset of Image Dataset

Sedan Template and First 5 labeled images

SUV Template and First 5 labeled image

Un-labeled images for either of the templates

(b) Labeling Results with two templates

Figure 8: Image Labeling Experiments on a Larger Dataset

Table 1 Performance evaluation of image labeling based on multi-level segmentation.

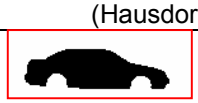
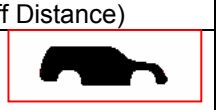
Image name		Car1	Car2	Car3	Car4	Face1	Face2	Face3	Face4
Single Level Segmentation	Number of Regions	14	12	21	12	13	9	12	18
	Number of valid combination tested in first search	4793	1363	50072	1264	2639	320	1166	49725
	Number of valid combination tested in subsequent search	10	83	5993	131	35	4	47	940
	Similarity (*)	5.6	8.4	11.2	5.8	0.9	0.7	0.5	0.6
Multi-level Segmentation	Segmentation level	2	1	2	10	2	3	2	10
	Number of regions	3	2	3	11	3	4	4	11
	Number of valid combinations at first search	7	3	6	735	7	12	12	824
	Number of valid combinations at subsequent searches	3	2	3	60	3	3	6	41
	Similarity (*)	3.7	7.6	11.7	5.1	0.9	0.7	0.5	0.5

* The similarities shown in the table are Hausdorff distances and histogram intersections for the car and face images respectively.

Table 2 Computational results of image labeling experiment on a larger dataset

	No. of Combinations tested for first 5 labeled images using **Sedan Template**	No. of Combinations tested for first 5 labeled images using **SUV Template**
Single Level Segmentation	71,865	67,312
Multi-level Segmentation	939	1069

Table 3 Similarity Measuring image labeling experiment on a larger dataset

Image No (As in Figure 8a).	Type	Visual Description	Similarity (Hausdorff Distance) 	
1	Sedan	Profile Shot	2.944	7.285
2	Sedan	Profile Shot	3.162	7.046
3	Sedan	Profile Shot	5.286	9.829
4	Sedan	Profile Shot	5.853	10.659
5	Sedan	Profile Shot	5.728	8.639
6	Sedan	Profile Shot	3.369	6.703
7	Sedan	Profile Shot, slightly rotated out of the image plane	7.624	10.499
8	Sedan	Profile Shot	3.100	6.622
9	Sedan	Profile Shot, slightly rotated out of the image plane	8.780	9.794
10	Sedan	Profile Shot. Occluded second car.	4.973	8.736
11	SUV	Profile Shot, somewhat rotated out of the image plane	9.009	7.487
12	SUV	Profile Shot	7.953	4.514
13	SUV	Profile Shot	6.817	3.701
14	SUV	Profile Shot, somewhat rotated in the image plane	11.985	11.656
15	SUV	Profile Shot	6.700	6.399
16 (Left Car)	Sedan	Profile Shot (Multiple Cars)	5.323	8.500
16 (Right Car)	Sedan	Profile Shot (Multiple Cars)	6.891	14.467
17	Sedan	Occluded Profile Shot	n/a	n/a
18	Sedan	Occluded Profile Shot	n/a	n/a
19	Sedan	Occluded Profile Shot	n/a	n/a
20	Sedan	Significant rotation out of the image plane	n/a	n/a