Object Recognition for Content-Based Image Retrieval

Yi Li and Linda G. Shapiro

Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
{shapiro,yi}@cs.washington.edu
http://www.cs.washington.edu/research/imagedatabase

Abstract. Content-based image retrieval (CBIR) refers to the retrieval of images according to their content, rather than through standard keyword retrieval techniques. Most of the early CBIR systems retrieved images based on global image features such as color histograms and texture statistics. These systems all had the philosophy of returning images that looked like a given example; they were not useful to users of commercial systems who wanted to find images containing specific objects. This paper addresses the need for object recognition in content-based image retrieval. It discusses the types of images features necessary for recognition of common objects in outdoor scenes and discusses three example systems that can recognize boats, vehicles, and buildings. It then goes on to discuss our proposed methodology for object recognition in terms of features called *abstract regions*, global representations for these features that can be used in learning, and a learning procedure that uses a hierarchy of support vector machines.

1 Introduction

Content-based image retrieval has become an important research area in computer vision as digital image collections are rapidly being created and made available to multitudes of users through the World Wide Web. [1] There are collections of images from art museums, medical institutes, and environmental agencies, to name a few. In the commercial sector, companies have been formed that are making large collections of photographic images of real-world scenes available to users who want them for illustrations in books, articles, advertisements, and other media meant for the public at large. The largest of these companies have collections of over a million digital images that are constantly growing bigger. Incredibly, the indexing of these images is all being done manually–a human indexer selects and inputs a set of keywords for each image. Each keyword can be

[1] This material is based upon work supported by the National Science Foundation under Grant No. IIS0097329. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

augmented by terms from a thesaurus that supplies synonyms and other terms that previous users have tried in searches that led to related images. Keywords can also be obtained from captions, but these are less reliable.

Content-based image retrieval research has produced a number of search engines that can retrieve images based on local or global features derived from color, texture, and simple shape information. The commercial image providers, for the most part, are not using these techniques. There are two main reasons for this:

1. Most CBIR systems require an example image and then retrieve similar images from their databases. Real users do not have example images; they start with an idea, not an image.

2. Most CBIR systems retrieve images based on low-level features, such as color and texture. Users want to retrieve images according to higher-level concepts such as scene classes (outdoor/indoor, city/country) and the objects present in the scene (people, tigers, cars, sunsets).

Thus the recognition of generic classes of objects and concepts is needed to provide automated indexing of images for content-based retrieval. In this paper, we suggest a methodology for object recognition through learning based on mid-level image features.

2 Related Literature

Object recognition is a major area of computer vision, but recognition of generic object classes is still an unsolved problem. While early work on recognition (e.g. the University of Massachusetts VISIONS System [12]) attempted to analyze complex natural scenes, the task was initially too difficult. Instead, research shifted to more practical domains with limited numbers of objects. Much of the important work in object recognition in the 1980s and 1990s was in the domain of industrial machine vision, where the objects to be recognized were specific industrial parts with fixed geometric models. In this domain, recognition refers to identifying an exact copy of a known 3D object, usually from the 2D projections of its detectable features, such as straight and curved line segments [4]. Objects to be recognized are represented by their visible features and by geometric invariants related to these features [10]. Once some of the features from an object are detected, the position and orientation parameters of the object are estimated, and its 3D geometric model is projected onto the image for a verification phase [13]. The geometric approach, for the most part, does not extend from single objects to classes of objects, especially not to classes of real-world objects that appear in general photographic images. However, the *feature-based approach* is an important object-recognition technique that is itself extendable to object classes.

In recent years, the computer vision community has started to tackle more general, more difficult recognition algorithms using a number of techniques that have been developed over the years. Techniques that use the appearance of an object in its images, instead of its 3D structure, are called *appearance-based* object recognition techniques [19][27][23]. Appearance-based techniques have been used to identify people by their faces and to match pictures of cars and other objects. The current limitations of these techniques are that they expect the image to consist of, or be limited to, the object in question and that this object must be presented from the same viewpoint as the images used to train the system (ie. front view of faces, side view of cars). Appearance-based techniques have been able to yield high recognition accuracy in limited domains.

Appearance-based techniques do not attempt to segment the image; this is both a strength and a weakness of the approach. *Region-based* techniques [2][26] do require presegmentation of the image into regions of interest. In most applications, the reliability of image segmentation techniques has been a problem for object recognition, but newer image segmentation algorithms [17][24] that use both color and texture can now partition an image into regions that, in many cases, can be identified as having the right colors and textural pattern to be a tiger or a zebra or some other object with a well-known color-texture signature. Related to this approach are algorithms that look for regions in color-texture space that correspond to particular materials, such as human flesh [8]. Such algorithms can be used with eye, nose, mouth recognizers to detect human faces or with constraints on region relationships to detect unclothed people. A different set of color criteria and spatial region relationships can be used to find horses [9]. People's faces have also been successfully detected using only gray-tone features and relying on heavily-trained neural net classifiers [20]. In fact, neural nets and support-vector machines have become an important tool in recognizing several different specific classes of imagery.

In the CBIR community, only a small number of researchers have worked on retrieval via object recognition and many of these efforts have been limited to a single class of object, such as people or horses. Some systems allow the user to sketch the shape of a desired class of object and retrieve images with similarly-shaped regions [5]. Recent systems are starting to embody general methods for object recognition and for concept recognition. For example, the Berkeley Digital Libraries group represents each object class as a hierarchy of image regions and their spatial relationships [9]. The work at Michigan State in concept recognition [28] uses a Bayesian classifier with lower-level features to classify different kinds of vacation images. A new and very promising approach to object classes [7] models objects classes as flexible configurations of parts, where the parts are merely square regions selected by an entropy-based feature detector [29], and also uses a Bayesian classifier for the final recognition task.

For the most part, generic object recognition efforts have been standalone. There is not yet a unified methodology for generic object class recognition or for concept class recognition. The development of such a methodology is the subject of our research.

3 Features for Object Class Recognition

In industrial machine vision, the main features used have been points, straight line segments, and to a smaller extent, curved line segments. In medical-image object recognition, intensity, texture, and shape of image regions are the main features. In content-based retrieval so far, the main features of interest have been the color and texture of image regions and the spatial relationships among them. Region shape has been used to a lesser extent, since it is less reliable for arbitrary views of 3D objects.

We work in the domain of outdoor scenes including city scenes, park scenes, and body of water scenes with such objects as sky, water, grass, trees, flowers, walkways, streets, buildings, fences, cars, trucks, buses, and boats. The object classes to be recognized require many different features for the recognition task. The major features of these object classes are their color, their texture, their structure, and their position in the image. Also useful in some cases will be region shape, such as ellipses for vehicle wheels or long thin rectangles for sailboat masts. Finally, some objects may be recognized on the basis of both their own features and those of their surroundings. We illustrate the variety of features that may prove useful with three examples from our work: boat recognition, vehicle recognition, and building recognition.

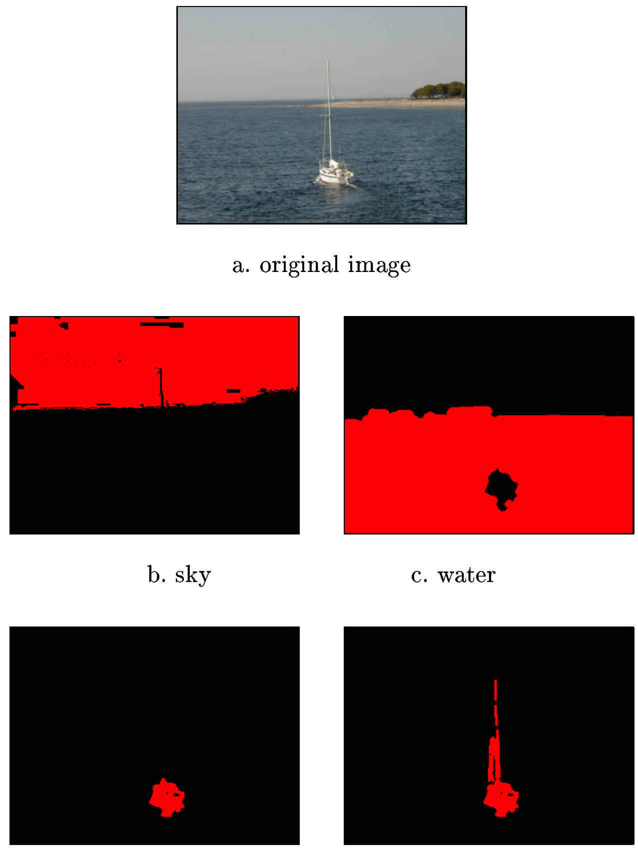
3.1 Features for Boat Recognition

Boats are examples of objects that can be recognized on the basis of both their own features and those of their surroundings. Figure 1 shows a very simple boat recognition system that we developed to illustrate the importance of context in object recognition. The algorithm for detecting boats in outdoor scenes is as follows.

1. First detect the sky. A region is sky if it a has blue color and a relatively flat texture.

2. Next look for water. A region is water if it has a blue color and a relatively rough texture. The properties of the region can be used to further verify the hypothesis. For example, a water region is usually at the bottom of an image and occupies a relatively large area.

3. Finally, look for a boat in the water. Boats have characteristic colors. For example, most pleasure boats are white, while commercial boats are often gray. Most importantly, boat regions can only be found within water regions.



a. original image

b. sky

c. water

d. boat

e. sailboat

Fig. 1. Recognition of boats using color regions, texture regions, and context.

To extend this approach to sailboats, the system can look for a vertical line segment above and adjacent to the boat region. This simple approach works quite well for finding possible boats, but it does not extend to other transportation objects, such as cars, trucks, and buses.

3.2 Features for Vehicle Recognition

There has been quite a lot of work in vehicle recognition, both feature-based and appearance-based. The eigenvector approach that was developed for face recognition has also been used for car recognition. In this case, the system must be trained on several different potential views of the vehicles to be recognized. Features used have included symmetry features for front and rear views and shape features (ie. parallelogram windows and elliptical wheels) for side views.

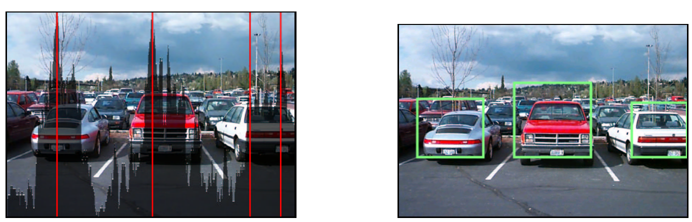
The symmetry approach [15] calculates a symmetry score for each image column by comparing edge pixels on each side and locates the central axes of the vehicles by finding the local maximum of the scores. Alternatively, for each image row, a score can be acquired for all the possible reflective points and all the possible symmetric lengths [31]. Summarizing the scores from all the rows and the peaks in the summation images determines the symmetric axes and the possible widths. The horizontal-edge-based approach [18] detects horizontal edges, forms a histogram by counting the horizontal edges on each image column, and selects the histogram peaks as the hypotheses of the central axes of vehicles. Figure 2 illustrates a vehicle recognition system that looks for front and rear views of cars and trucks. The vehicle detection system operates as follows:

1. The system locates the central axes of vehicles using either the symmetry-based approaches or the horizontal-edge-based approach. More reliable locations can be chosen by voting from just one approach.

2. The length of horizontal lines and the peak ranges of the maxima in the last step are used to determine the width of the vehicles.

3. The top and bottom boundaries of the vehicles are determined by finding horizontal line clusters. The boundaries can be verified by under-vehicle shadows.

The system is not very robust; partial views of vehicles and those that are far away will not be detected.



a. horizontal-edge histogram   b. detected vehicles

Fig. 2. Recognition of front and rear views of vehicles using symmetry features and horizontal line segments.

3.3 Features for Building Recognition

Many man-made objects are too complex for the above features. Such objects as buildings, houses, buses, and fences, for example, are not segmentable through color or texture alone and have many line segments rather than one or two important ones. What they do have is a very regular structure, consisting of multiple line segments in one or two major orientations and usually just one or two dominant colors. We have developed a building recognition system [16] that uses *structure features*. These features are obtained as follows:

1. Apply the Canny edge detector [1] and ORT line detector [6] to extract line segments from the image.
2. For each line segment, compute its orientation and its color pairs (pairs of colors for which the first is on one side and the second on the other side of the line segment).
3. Cluster the line segments according to their color pairs, to obtain a set of *color-consistent* line clusters.
4. Within the color-consistent clusters, cluster the line segments according to their orientations to obtain a set of color-consistent *orientation-consistent* line clusters.
5. Within the orientation-consistent clusters, cluster the line segments according to their positions in the image to obtain a final set of *consistent line clusters.*

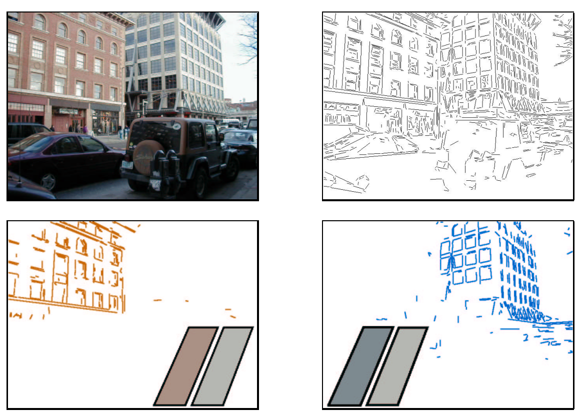


Fig. 3. (top left) Original image. (top right) Line segments. (bottom) Color-consistent line clusters.

Color-Consistent Line Clusters To reduce the complexity of obtaining color-consistent line clusters, we first classify each pixel of the image as one of several dominant colors, using the Gong color clustering algorithm [11] and the CIEL*a*b* color space. Then each line segment is assigned one or more color pairs consisting of one dominant color from its left region and one from its right region, based on a small window of analysis. The line segments are grouped into color-consistent line clusters based on these color pairs. Figure 3 illustrates the process of constructing the color-consistent line clusters. The main color pair of the left building in Figure 3 is (tan,gray), while the main color pair of the right building is (grayblue,gray). The two color clusters (bottom row) also contain spurious segments from other objects.

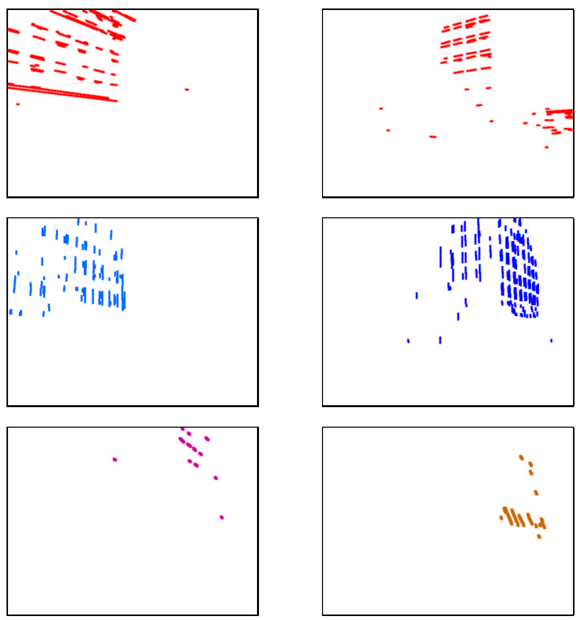


Fig. 4. Orientation-consistent line clusters obtained from the color-consistent line clusters shown in Figure 3. The results are final orientation-consistent clusters using both orientation and perspective information with small clusters removed.

Orientation-Consistent Line Clusters For every color-consistent line cluster, the orientation feature of the line segments can be used to further classify them. We would like to assign the parallel segments of an object to exactly one orientation-consistent line cluster. Because of the effect of perspective projection, the parallel lines on an object may not be parallel in the image, but will converge to a single point. Because of this, we use two steps to achieve our objective: first, roughly classify the segments according to their orientation in the image, and second, decide whether they are parallel to each other or they converge to a vanishing point in the image. Finding the roughly orientation-consistent line clusters is achieved through a simple clustering algorithm that finds the peaks in the orientation histogram and assigns each line segment to the cluster associated with its closest peak. After the roughly-orientation-consistent line clusters are obtained, the perspective information is used as a key both to decide whether the segments in a line cluster are consistent and to filter out the "noise" lines. Each of the two color clusters in Figure 3 produced several orientation-consistent clusters as shown in Figure 4.

Spatially-Consistent Line Clusters After constructing the consistent line clusters using color and orientation features, the resultant clusters may still have some segments from different physical entities. To rule out such segments, spatial clustering is performed using both vertical and horizontal position histograms. First, the line segments in a cluster are projected to the y-axis to create a vertical position histogram, which can be segmented into groups of y-positions that yield vertical position clusters. Then, the line segments of each vertical position cluster are projected to the x-axis to create a horizontal position histogram whose segmentation produces horizontal position clusters. The line segments in the resultant spatially-consistent line clusters are close to each other, both vertically and horizontally, in the image. The application of color-consistent clustering followed by orientation-consistent clustering followed by spatially-consistent clustering yields the set of consistent line clusters that are used to detect buildings or other line-segment-rich structures. Figure 5 shows two spatially-consistent line clusters which came from the single orientation-consistent line cluster in the top-right position of Figure 4. The cluster has been divided into the line segments from a building and those from an automobile.

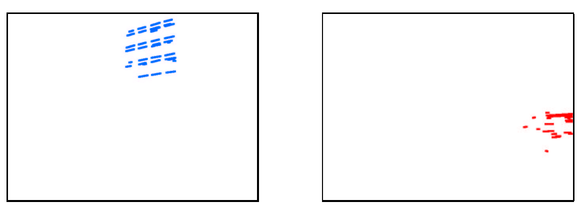


Fig. 5. Two spatially-consistent line clusters obtained from the single orientation-consistent line cluster shown in Figure 4 (top-right image).

Recognizing Buildings Once the consistent line clusters have all been constructed, they can be used to detect objects, such as buildings. We use two criteria to detect buildings: the interrelationships of the consistent line clusters detect structure-preserving or *junction-rich buildings*, and their intra-relationships can be used to detect the simple-structured or *overlap-rich buildings*. These two situations are shown in Figure 6 and discussed below. The location of a building can be estimated from the position of its corresponding line clusters.

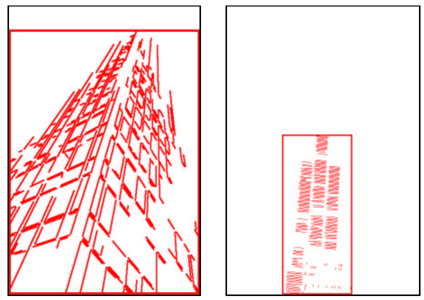


Fig. 6. (left) Interrelationship criterion. (right) Intra-relationship criterion.

Interrelationship Criterion Because many line segments on buildings are from windows and doors, there will be many intersections. Two line segments are classified as intersecting if when they are extended, the intercept point of the two virtual lines is on one of the line segments or is close to one of their end points. For every line cluster pair (cl_i, cl_j), the interrelationship between the two line clusters is considered to decide whether this cluster pair is a hint of the existence of buildings. In order for the pair to be a qualified hint, its line segments should form enough intersections. This can be represented by $Ninter_{ij}$, the total number of lines in cl_i that intersect some line in cl_j. For each cluster, we can then define $Ninter_i$, the maximum number of intersecting lines of cl_i over all other clusters, and finally, its normalized version $Norm\_inter_i$, normalized by the total number of lines in the cluster.

$$Ninter_{ij} = |\{l_1 \in cl_i \mid \exists l_2 \in cl_j,\ l_1\ intersects\ l_2\}|$$

$$Ninter_i = max_j\{Ninter_{ij}|\ i \neq j\ \}$$

$$Norm\_inter_i = \frac{Ninter_i}{|cl_i|}$$

Intra-relationship Criterion Due to the many different appearances of buildings and the different distances at which the images were taken, some buildings do not produce many junctions on the image, but they do have many overlapping line segments. The intra-relationship criterion is used to examine how many lines heavily overlap in a line cluster cl. For a line l, $Nol(l) = |\{l'|l'\ overlaps\ l\}|$ is the number of lines in cl that overlap with l. If $Nol(l)$ is large enough (greater than a learned threshold T_{ho}), then l is a heavily overlapped line. The number of heavily overlapped lines is another hint of the existence of buildings. Similar to the interrelationship criterion, the intra-relationship criterion is defined by features: $Nintra_i$, the number of heavily overlapped lines in the line cluster cl_i, and $Norm\_intra_i$, normalized by the total number of lines in the cluster.

$$Nintra_i = |\{l | l \in cl_i \ and \ Nol(l) \geq T_{ho}\}|$$
$$Norm\_intra_i = \frac{Nintra_i}{|cl_i|}$$

Evaluation of Building Detection Our consistent-line-cluster (CLC) features can be used for two different tasks: 1) content-based image retrieval and 2) object recognition. For content-based image retrieval, we built a simple decision-tree classifier. The feature vectors for this task were designed to convert the local features for the separate clusters to a global histogram as follows:

$$Ninter(k) = max\{Norm\_inter_i | Ninter_i \geq k\}$$
$$Nintra(k) = max\{Norm\_intra_i | Nintra_i \geq k\}$$

for $k = 1$ to max_k. With the feature vectors

$$(Ninter(1), \ldots, Ninter(max_k),$$
$$Nintra(1), \ldots, Nintra(max_k))$$

and max_k set to 64, we used the C4.5 package to generate simple decision-tree classifiers. Our test database of 977 images was obtained from two online image databases: creatas.com and freefoto.com. We selected 336 building images and 641 nonbuilding images and ran a set of cross-validation experiments in each of which 90% of the images were used as the training set and the other 10% as the test set. The average error (false positives plus false negatives) over the set of cross-validation experiments was 5.8%. (Note that our algorithms were developed on a completely independent database of our own images of campus, city, and landscape scenes.)

buildings buildings cactus

Fig. 7. Sample images from the online test set.

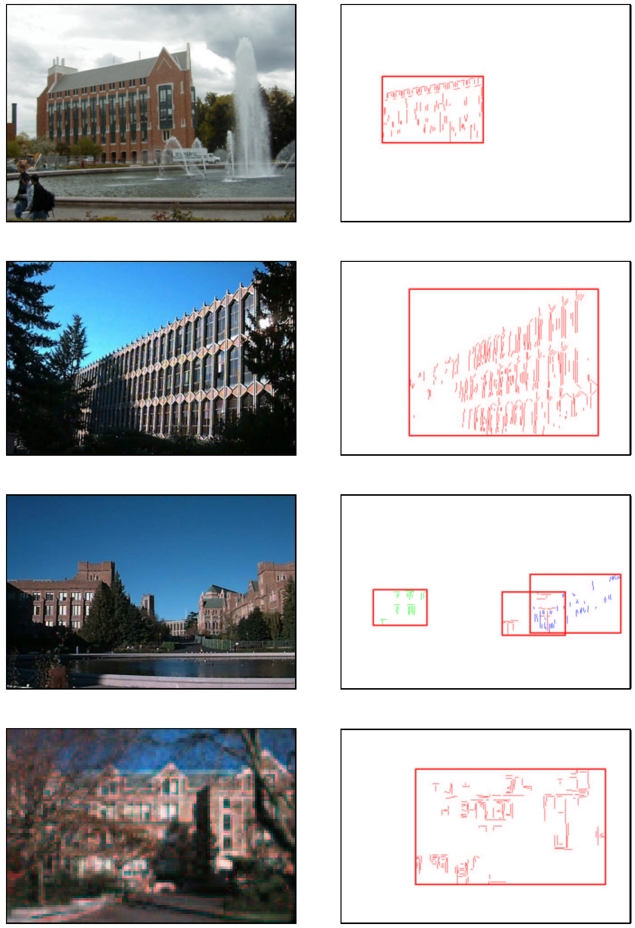


Fig. 8. Some correct classifications.

There are three CBIR methods that seem most related to our own: Iqbal and Aggarwals's approach [14] to building recognition using perceptual grouping (rectangles), Zhou, Rui, and Huang's water-filling algorithm [30] for extracting edge-map features, and Vailaya, Jain, and Zhang's edge-direction-histogram (EDH) features [28] for classifying city vs. landscape images. Because the emphasis in [14] and [30] was quite different from our own and since several early reviewers of our paper suggested comparing our features to the EDH features, we implemented EDH and tested it on classifying building vs. nonbuilidng images, using C4.5 and cross validation as above. On the same test set, the EDH method had an average error of 16.5% compared to CLC's 5.8%. Note that this test data set (see Figure 7 for examples) is a fairly difficult one with the building images taken from many different viewpoints and the nonbuilding images containing many vertical line segments. We found that EDH worked very well on images with many vertical lines and less well on those without vertical lines. Our CLC features are not sensitive to a particular orientation and can also handle perspective.

The CLC features can also be used for object recognition and approximate localization, so that the presence and locations of objects in images can be detected and indexed for future retrievals. For object recognition, we tested the algorithm on 97 well-patterned buildings, 44 non-well-patterned buildings, 16 non-patterned non-buildings, and 25 patterned non-buildings in a local test set, specifically acquired to control these experiments. The results were 0% error for well-patterned buildings, 4.5% error for non-well-patterned buildings, 6.2% error for non-patterned non-buildings, and 100% error for the patterned non-buildings, which were objects like faculty mailboxes, buses, and fences, all selected to try to fool the algorithm (and they all did!). Additional features (context) will be required to distinguish these similarly structured objects from buildings. Figure 8 shows some correct classification/location results on this image set. The first and second row images were considered to contain well-patterned buildings, while the third and fourth row images, which are more difficult, were considered to be non-well-patterned.

Two of the misclassifications are shown in Figure 9. The reasons for false negatives are the lack of enough patterns and the pattern interference from other objects, for example, trees. Some patterns from objects other than buildings, such as trees and bridges, are recognized as buildings in the false positives.

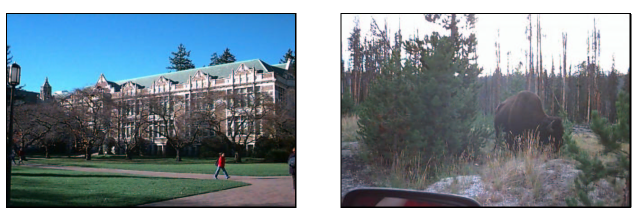


Fig. 9. False negative (left) and false positive (right).

Two images from the patterned non-buildings are shown in Figure 10. Although they are currently false positives, they also show the potential use for consistent line clusters, along with other features, for recognizing additional man-made objects.

Fig. 10. Two false positives that show the potential for using consistent line clusters to find additional man-made objects.

4 Abstract Regions and Learning

We are developing a new methodology for object recognition in content-based image retrieval. Our methodology has three main parts:

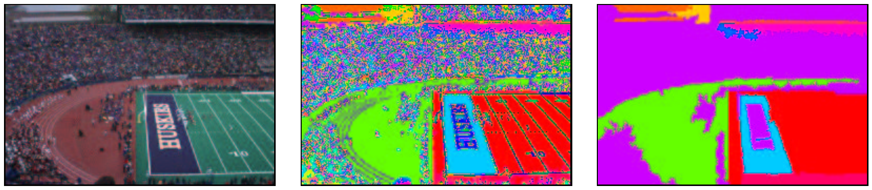
1. Select a set of mid-level features that have multiple attributes for recognition and design a unified representation for them.

2. Develop methods for encoding complex features into feature vectors that can be used by general-purpose classifiers.

3. Design a learning procedure for automating the development of classifiers for new objects.

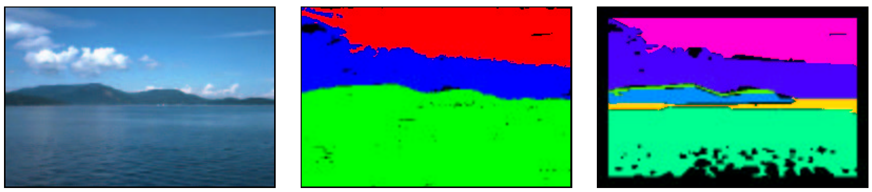
The unified representation we have designed is called the *abstract region* representation. The idea is that all features will be regions, each with its own set of attribures, but with a common representation. The regions we are using to start our work are color regions, texture regions, and structure regions. Color regions are produced by a two-step procedure. The first step is color clustering using a variant of the K-means algorithm on the original color images represented in the HSI color system. The second step is a merging procedure that merges multiple tiny regions into large ones. Figure 11 illustrates this process on a football image in which the K-means algorithm produced hundreds of tiny regions for the multi-colored crowd, and the merging process merged them into a single region.

Our texture regions come from a color-guided texture segmentation process. Color segmentation is first performed using the K-means algorithm. Next, pairs of regions are merged if after a dilation they overlap by more than 50%. Each of the merged regions is segmented using the same clustering algorithm on the Gabor texture coefficients. Figure 12 illustrates the texture segmentation process. Structure regions come from the features we developed for building recognition. They are polygons containing one or more overlapping consistent-line structures.



Original Color Merged

Fig. 11. Illustration of the merging of tiny color regions.



Original Image | Color Segmentation | Texture Segmentation

Fig. 12. Our texture segmentation is color-guided; it is performed on the regions of an initial color segmentation.

Figure 13 illustrates the abstract regions for several representative images. The first image is of a large campus building at the University of Washington. Regions such as the sky, the concrete, and the large brick section of the building show up as large homogeneous regions in both the color segmentation and the texture segmentation. The windowed part of the building breaks up into many regions for both the color and the texture segmentations[2], but it becomes a single region in the structure image. The structure-finder also captures a small amount of structure at the left side of the image. The second image (park) is segmented into several large regions in both color and texture. The green trees merge into the green grass on the right side in the color image, but the texture image separates them. No structure was found. In the remaining four images (sailboat, house, building with cherry trees, and flowers in front of a house) both the color and texture segmentations provide some useful regions that will help to identify the sky, trees, flowers, lawn, water, and sailboat; the sailboat, house, pieces of building, and pieces of house are captured in structure regions. It is clear that no one feature type alone is sufficient to identify the objects.

[2] The black regions are areas where there were many small regions, which have been discarded as not useful.

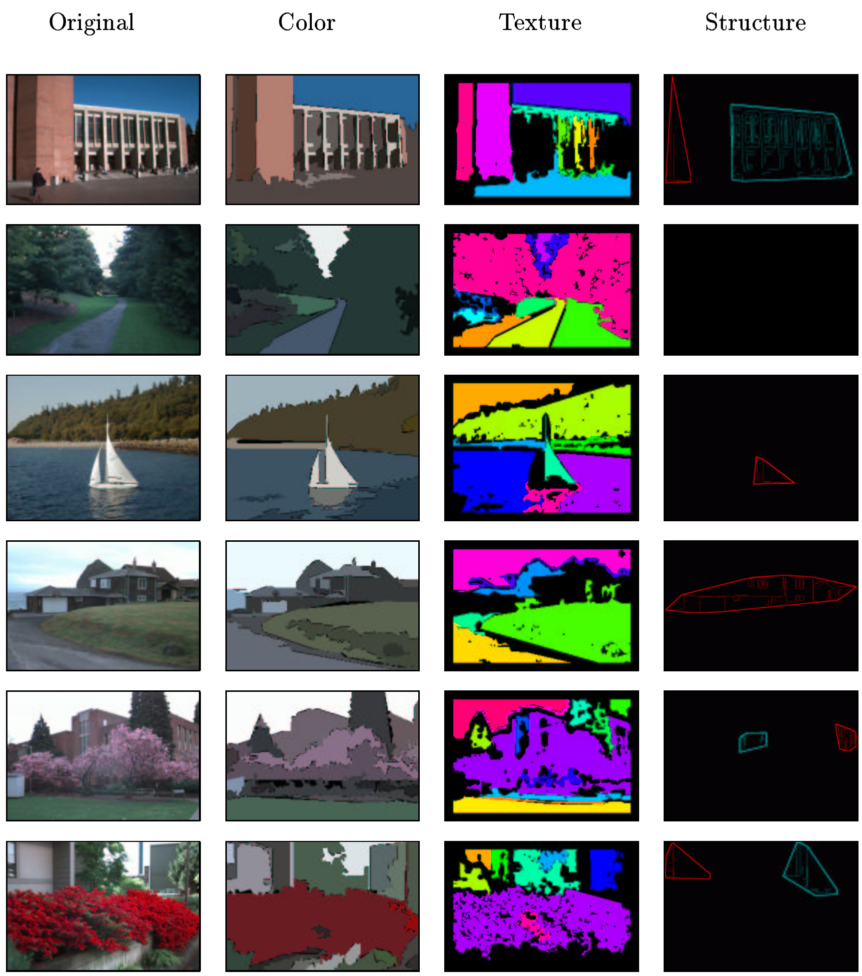


Fig. 13. The abstract regions constructed from a set of representative images using color clustering, color-guided texture clustering, and consistent-line-segment clustering.

The features described above are all examples of mid-level features. Abstract regions should also be able to handle other features that we have used or will use. For example, line segments can be represented as long, thin regions with length, orientation, and position. Symmetry features can be represented by regions that consist of the axes of symmetry and have statistical features, such as the mean width of the symmetric entity and the variance. Once we have these unified features, we still need a method for converting all the features in an image to a representation suitable for learning. We are investigating two main approaches to this representation: iconic and symbolic.

The iconic representation converts the abstract regions to labeled images: one for color regions, one for texture regions, and one for structure regions. Only the most significant regions will be chosen for this class. The chosen regions from the training data set will be clustered according to their attributes, producing a set of labeled color clusters, a set of labeled texture clusters, and a set of labeled structure clusters. The labels will represent the attributes of their clusters. For example, in the color clusters, we would expect one or more green clusters that may come about from trees, grass, or bushes. In the texture clusters, we will get separate labels for different types of texture, such as brick texture, tree texture, or ocean texture. The structure clusters will vary as to the color pairs from which they are derived and the orientations and densities of the line segments.

The symbolic representation is a data structure that summarizes the abstract regions in an image. We envision a histogram structure that keeps track of the sizes and locations of the nontrivial abstract regions in bins spanning their attribute spaces. The clustering idea suggested above may also be used in this approach. Both the iconic and symbolic image representations produce large feature vectors that can become the input to learning procedures.

The learning aspect of this work is in an early stage of development. In Ruiz-Correa's 3D shape recognition work [22], we have developed a new learning paradigm: a three-stage classification process that constructs, identifies, and labels shape regions in a 3D mesh and classifies shapes based on the types of regions and their spatial configuration. One approach we intend to try is a variant of this approach that uses our abstract regions in place of the 3D shape regions. We will also experiment with more conventional classification techniques and with hierarchical multiple classifiers [3].

5 Conclusions and Future Work

We have presented a methodology for object recognition for content-based image retrieval in the domain of color images of natural scenes containing both man-made and nature objects. We first investigated the mid-level features necessary for recognizing common objects. We found that many common objects, such as sky, water, trees, flowers, office buildings, houses, and vehicles can be recognized from a combination of mid-level features including color regions, texture regions, structure regions, major line segments, and symmetry features and their spatial arrangements. We developed systems to recognize boats, vehicles, and buildings using these features.

From these systems we designed a set of abstract region features to be used in systems that learn to recognize common objects from example images containing them. We also developed two different image representations that can capture the abstract regions of an image and their spatial relationships as a global feature vector that can be used in learning. Our next task is to apply learning algorithms to this data in order to test our hypothesis that the procedures for recognizing common objects can be learned from these representations.

References

1. J. Canny: A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, pp. 679-698, 1986.
2. C. Carson, S. Belongie, H. Greenspan, J. Malik: Region-based Image Querying, Proceedings of the 1997 IEEE Workshop on Content-Based Accesss of Image and Video Libraryies, pp. 42-49, June 1997.
3. Y. Chou and L. G. Shapiro: A Hierarchical Multiple Classifier Learning Algorithm, Proceedings of the International Conference on Pattern Recognition, 2000, Vol. 2, pp. 152-155.
4. Costa, M. S. and L. G. Shapiro, "3D Object Recognition and Pose with Relational Indexing," *Computer Vision and Image Understanding*, Vol. 79, No. 3, 2000, pp. 364-407.
5. A. Del Bimbo, P. Pala, S. Santini: Visual image retrieval by elastic deformation of object sketches, IEEE Symposium on Visual Languages, pp. 216-223, 1994.
6. A. Etamadi: Robust Segmentation of Edge Data, Proceedings of the IEE Image Processing Conference, 1992.
7. R. Fergus, P. Perona, and A. Zisserman: Object-Class Recognition by Unsupervised Scale-Invariant Learning, Proceeeings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2003, pp. 264-271.
8. M. M. Fleck, D. A. Forsyth, and C. Pregler: Finding Naked People, Proceedings of the European Conference on Computer Vision, Springer-Verlag, 1996, pp. 593-602.
9. D. Forsyth and M. Fleck: Body Plans, IEEE Society Conference on Computer Vision and Pattern Recognition, pp. 678-683, 1997.
10. D. Forsyth, J. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell: Invariant Descriptors for 3-D Object Recognition and Pose, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 10, 1991, pp. 971-991.
11. Y. Gong: Advancing Content-Based Image Retrieval by Exploiting Image Color and Region Features, Multimedia Systems, Vol. 7, pp. 449-457, 1999.
12. A. R. Hanson and E. M. Riseman: VISIONS: A Computer System for Interpreting Scenes, Computer Vision Systems, A. Hanson and E. Riseman, eds., New York, Academic Press, 1978, pp. 303-333.
13. D. P. Huttenlocher and S. Ullman: Recognizing Solid Objects by Alignment with an Image, International Journal of Computer Vision, Vol. 5, No. 2, 1990, pp. 195-212.

14. Q. Iqbal and J. K. Aggarwal: Applying Perceptual Grouping to Content-Based Image Retrieval: Building Images, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 42-48, June 23-25, 1999.
15. A. Kuehnle: Symmetry-based recognition of vehicle rears, Pattern Recognition Letters, vol.12, no.4, 1991.
16. Y. Li and L. G. Shapiro: Consistent Line Clusters for Building Recognition in CBIR, Proceedings of the International Conference on Pattern Recognition, 2002.
17. W. Y. Ma and B. S. Manjunath: NETRA: A Toolbox for Navigating Large Image Databases, Proceedings of the IEEE International Conference on Image Processing, 1997.
18. N. D. Matthews, P. E. An, D. Charnley, and C. J. Harris: Vehicle detection and recognition in greyscale imagery, Intelligent Autonomous Vehicles, pp. 1-6, 1995.
19. H. Murase and S. K. Nayar: Visual Learning of Object Models from Appearance, International Journal of Computer Vision, 1992.
20. H. Rowley, S. Baluja, and T. Kanade: Neural Network-Based Face Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 1, 1998, pp. 23-38.
21. Y. Rui, T. Huang, and S. Mehrotra: Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval, Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries, pp. 67–74, 1997.
22. S. Ruiz-Correa, L. G. Shapiro, and M. Meilă: A New Paradigm for Recognizing 3-D Object Shapes from Range Data, to appear in Proceedings of the International Conference on Computer Vision, October, 2003.
23. S. Sclaroff, A. Pentland: Object Recognition and Categorization using Modal Matching, Proc. of 2nd CAD-Based Vision Workshop, pp. 258-265, 1994.
24. J. Shi and J. Malik: Normalized Cuts and Image Segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 731-737.
25. J.R. Smith and S.-F. Chang: VisualSEEK: A Fully Automated Content-Based Image Query System, ACM Multimedia, pp. 87-98, November, 1996.
26. J. R. Smith and C. S. Li: Image Classification and Querying Using Composite Region Templates, Computer Vision and Image Understanding: Special Issue on Content-Based Access of Image and Video Libraries, Vol. 75, Nos. 1-2, 1999, pp. 165-174.
27. M. Turk and A. Pentland: Eigenfaces for Recognition, Journal of Cognitive Neuroscience, 1991.
28. A. Vailaya, A. K. Jain and H.-J. Zhang: On Image Classification: City Images vs. Landscapes, Pattern Recognition, Vol. 31, pp 1921-1936, 1998.
29. M. Weber, M. Welling, and P. Perona: Unsupervised Learning of Models for Recognition, Proceedings of the European Conference on Computer Vision, 2000, pp. 18-32.
30. S. X. Zhou, Y. Rui, and T. S. Huang: Water-filling Algorithm: A Novel Way for Image Feature Extraction Based on Edge Maps, Proc. IEEE Int. Conf. on Image Proc., 1999.
31. T. Zielke, M. Brauckmann, and W. Von Seelen: Intensity and edge-based symmetry detection with an application to car-following, CVGIP: Image Understanding, vol.58, no.2, 1993