Combining spanning trees and normalized cuts for Internet retrieval

Sharat Chandran[a] (sharat@acm.org) & Abhishek Ranjan[b] (abhishekranjan@iitb.ac.in)

[a] Computer Science Department, Indian Institute of Technology, Bombay &
Center for Automation Research, University of Maryland.

[b]Computer Science Department, Indian Institute of Technology, Bombay.

ABSTRACT

Graph based approaches succeed in producing *meaningful* regions in images when they are to be stored as entities in a database for *content-based retrieval*. Despite controlling various parameters, the bottom-up approach[1] produces too many segments for an Internet search retrieval scheme. The top-down scheme[2] can be adjusted for wide area searches, but has a high computational cost.

In this work we combine the two approaches and retain the advantages of both approaches. The key idea is to use local approach for reducing the size of the problem that is fed to the normalized cut approach. Our algorithm runs in $O(n \log n)$ time.

Keywords: Normalized cuts, clustering, Internet retrieval

1. INTRODUCTION

The amount of images, audio, and video data in easily accessible user spaces such as personal computers, digital libraries, multimedia databases, and the Internet has exploded. As a result the problem of retrieving images and videos by *content* (as opposed to retrieval based on text annotation) has been increasingly studied by several researchers in the last few years. In this context, it is interesting to note that the first commercial text based search engine (Yahoo) for the World Wide Web (1991–1993) appeared in 1994. However, a decade later, we find existing image based search engines less than satisfying (Google's image based search debuted in 2002, and is essentially text based — See Figure 1).

Figure 1. A query using Google's Image Search for the apple image on the top left produces a variety of pictorially irrelevant images. While context is certainly important in search, the focus of this work is in the image content.

Our focus in this paper, like those of many others, is on retrieving images based on intuitive pictorial concepts rather than based on strings appearing in accompanying text based articles. In retrieving images by content, one cannot overestimate the use of appropriate features such as color, texture, and shape. However, it has also become increasingly evident that the decomposition of images into regions is critical for useful results. Once regions are produced, there are several algorithms and systems that will output images and video sequences that match user search terms (which are now sketch or image based).

1.1. Our proposal

There are two main ideas in this paper

- We use a graph based paradigm (specifically normalized cut[2] (N-cut)) in creating meaningful clusters. While there are several schools on how to produce regions, the graph theoretic approach has gained prominence in the recent[2–4] literature. These graph theoretical approaches combine concepts such as normalized cut and spectral graph theory and provide what one might call *natural* segmentation.

  On the other hand, one reason for the lack of universal popularity of graph algorithms despite faster computers, and use of clever numerical techniques, is that these algorithms run slowly, and still remain beyond mainstream Internet usage. Even making some assumptions on the sparsity of certain matrices, the core routine in the N-cut algorithm takes $O(n^{1.5})$ time for a graph with n nodes. A segmentation scheme across the Internet requires repeated use of this core super linear algorithm, and therefore the time requirements can make the approach intractable. Our approach of combining minimum spanning trees, and the normalized cut makes the graph algorithm tractable.

- A related assumption in this work is that the image database is across a wide area network. This poses additional challenges that could be considered in more depth than is reported in the literature.

  - Network congestion is common across the Internet, especially for popular sites or events. A progressive refinement strategy would be useful in which the user first gets approximate results.
  - User profiles are common across the Internet. It is important for relevance feedback (for example[5]) to be incorporated.
  - Speedy response is paramount. While the goal of every system is to be fast, in searches across the Internet, users have now accepted the appearance of "wrong results."
  - Input sources that serve as cues for search need not live on local resources (for example, due to copyright restrictions). It might be useful to point publicly accessible queries from sources across the Internet.

We discuss more details of our approach in the rest of this paper. which is organized as follows. After a discussion of previous work in Section 2, we give details of our approach in Section 3. This section first introduces the two main "ingredients" in graph partitioning and shows how they can be pipelined to make an overall scheme tractable. Sample results are shown in Section 4 and we end with some concluding remarks in the last section.

2. PREVIOUS WORK

There has been a lot of interest in content based image retrieval in the recent years. The first wave of research incorporated color histogram as features. Image feature vector indexing techniques have been implemented in various systems like QBIC,[6] Photobook,[7] and WBIIS[8] with varying degrees of success. One problem with the color histogram approach is that histograms do not contain the shape, location, or texture information. Two images having the same color histogram may differ widely in shape and image semantics.

An alternative method is to use[8–10] wavelets for the image signatures. Since wavelets capture shape and location to some extent, some problems are eliminated. However, these systems usually consider the lower frequencies in the image. Since texture is known to have high frequencies,[11] these systems show deteriorated performance in some cases.

Significantly, since a single signature is calculated for an image, the system suffers when, for instance, two images contain similar objects that are spatially varied. Most of the above systems have trouble in handling rotated, translated, or scaled versions of content.

The more general approach is to reconsider the classic segmentation problem in computer vision, and retrieve similar images based on regions within a query image. Blobworld[12] is a system which does segmentation based on high level features of the image. The user is asked to select a blob on the image and a few other parameters (such as the importance of background) to assign the weightages for each of the feature vectors. Simplicity[13] is another segmentation based system. It uses the "Integrated Region Matching"[14] approach to compute the distance between query image and target image. WindSurf[15] is another wavelet based system which does image segmentation using a k-means clustering algorithm. It uses both the low level and the high level wavelet coefficients, thus using texture in the process of segmentation.

These systems underscore the importance of working with regions in retrieving images similar to a query image. As a double-edged sword, improper segmentation leads the implementation astray in many cases as can be seen by trying out the systems.

The 2D string matching work[16] shows an attempt to automate object recognition. Another recent system developed at the University of Massachusetts[17] uses an automatic iterative segmentation algorithm with domain knowledge-driven feedback. The system is used to index flower patent images using domain knowledge. The system is shown to perform reasonably well on the flower domain. However, it is difficult to extend the system to perform queries on a generalized domain.

A largest common subgraph (LCS) technique[18] for video indexing and similarity has been proposed. However, the system considers an exact LCS algorithm to find out the similarity between two images. This results in an exponential time for retrieving similar videos.

3. OUR APPROACH

Clustering is one of the best known problems in computer vision, and has been vigorously addressed for the last 30 years. Graph theory being a well studied branch of computer science has been exploited[19] for various segmentation purposes. Specifically, vertices in the graph contain features such as color, texture, and motion profiles. Edges might correspond to how these vertices are associated with each other; a strong link might suggest that these vertices are similar. Graph approaches may be classified in two approaches.

In the *top down* approach, a minimal cut is made to partition the graph into two. Significantly, further segmentation is achieved recursively. In the *bottom up* approach clusters are created, possibly in parallel, at several places and each cluster is represented by an appropriate data structure such as the minimum spanning tree. The minimal cut between pairwise clusters is considered, but this time to decide whether clusters should be agglomerated or not.

Shi and Malik[2] propose the notion of *normalized* cuts. This technique produces good results in segmentation but is expensive. As an example, consider Figure 2. Unfilled circles represent the foreground and occupy a small but significant portion of the image. Completely filled circles represent the background. A large portion of the data is occupied by noisy data and is shown using shaded circles. The normalized cut method succeeds in creating an intuitive segmentation. A bottom approach correctly produces three segments but the higher level process is confused as to which areas are significant for Internet retrieval.

In this section, we present the salient technical parts of the two approaches (dubbed as Algorithm P and Algorithm N-cut) we have adopted. Algorithm-P[20] is an updated version of the bottom-up approach.[1] We skip many details that can be found in the original papers.

3.1. Algorithm-P

Given a set V of elements (for example, shots) to be segmented, the goal is to find a partition, or *segmentation* $S = \{C_1, C_2, \ldots, C_p\}$. We denote by $D(C_i, C_j)$ a pairwise region comparison Boolean function that judges whether or not there is evidence for a boundary between two components C_i and C_j. S is said to be *too fine* when there is some pair of regions C_1 and C_2 for which $D(C_1, C_2)$ is false. Given two segmentations S and T of

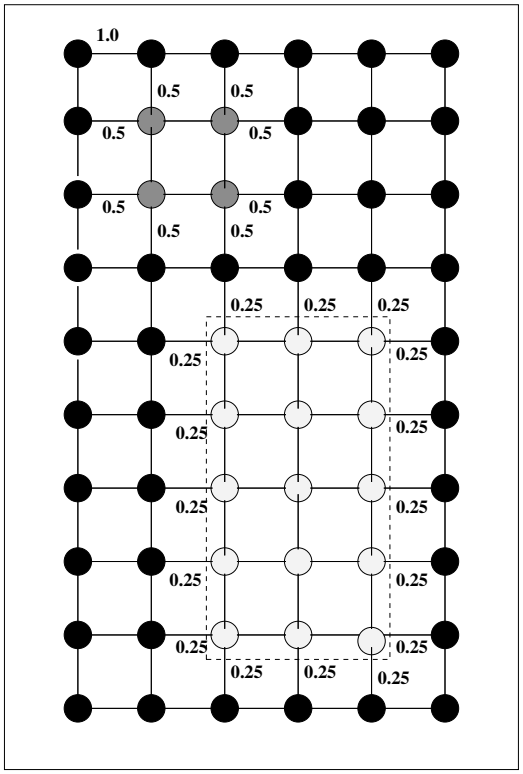

Figure 2. The normalized cut method is expensive, but can recover the intuitive foreground suggested by unfilled circles.

V, T is said to be a proper refinement of S when $\forall C \in T$, $\exists C' \in S$ such that $C \subset C'$. S is said to be *too coarse* when there exists a proper refinement of S that is not too fine.

A segmentation is *good* if it is neither too fine nor too coarse. There are several (different) good segmentations for the same image, and the nature of D dictates some of them. We use the scheme in[20] which is elegant (because it produces a good segmentation) and fast (it runs in almost linear ($O(n \log n)$) time). The main disadvantage with this algorithm is that the goal is to produce a complete segmentation; there is no easy way to produce a good segmentation which contain meaningful data to be used as queried for searching.

3.2. Algorithm N-cut

Let $G(V, E)$ be a graph such as the shot similarity graph with $|V| = n$, and $w(i, j)$ denote the weight of the edge joining v_i, v_j. The *normalized cut* is defined as

$$Ncut(C_i, C_j) = \frac{cut(C_i, C_j)}{assoc(C_i, V)} + \frac{cut(C_i, C_j)}{assoc(C_j, V)} \tag{1}$$

where $cut(A, B) = \sum_{v_i \in A, v_j \in B} w(i, j)$ and $assoc(X, V) = \sum_{v_i \in X, v_j \in V} w(i, j)$

The quantity $Ncut$ denotes how strongly connected nodes of C_i are among themselves as compared to the connection to another component C_j. In this sense it is similar conceptually with Algorithm-P. However the lower $Ncut$ is, the better the segmentation so that it makes sense to minimize this quantity globally across the entire graph. As mentioned earlier, this global minimization has a price. It requires quadratic space complexity which becomes too large for video sequences. Also, even with a sparsity assumption of certain matrices, it runs in $O(n^{1.5})$ time. This assumption was valid in the context of the original segmentation algorithm, but may not necessarily be valid in all cases.

3.3. Central Idea

The key idea is to take advantage of the speed of Algorithm-P and the top-down global nature of Algorithm N-cut which produces the foreground data. If we somehow feed an input of size $\sqrt{n}$ to Algorithm N-cut, it will result in manageable space complexity of $O(n)$. It will also result in one call running in time $n^{0.75}$. As a result, the net algorithm runs in $O(n \log n)$ time.

The delicate aspect of this pipelining procedure is to feed the *proper* "super pixels" to Algorithm N-cut. Recall further that we cannot predict the size of the output segmentation in Algorithm-P. It may be $O(\sqrt{n})$ which is acceptable, or it might be $o(\sqrt{n})$ which is also acceptable. If, however, the output size is $\Omega(\sqrt{n})$ we are in trouble! This requires further exploration of Algorithm-P as described below.

3.3.1. Some Details

Algorithm-P produces only clusters, but we return at the end of the first stage in the algorithm a set of clusters *and* edges between clusters. This edge is precisely the edge which causes disagreement between cluster and was thrown away by Algorithm-P. When the cluster being grown currently encounters a node v which cannot be merged, a link is created from this cluster to the cluster which contains v.

We assign a weight to this link which is an increasing function of similarity between the two clusters. This process continues till the whole image is segmented. The algorithm forms a set E' of links. This step takes $O(n \log n)$ time. An example illustrating the idea of this step has been shown in Figure 3.

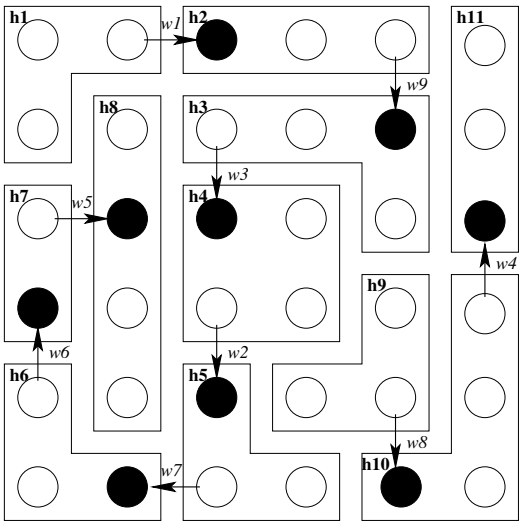

Figure 3. Illustration of the first step in our algorithm. Clusters are shown as polygons and linking nodes are colored in black. Annotated arrows show the weighted links between two clusters.

If the number of clusters obtained in the previous step is very large then we perform cluster merging to reduce the number of clusters. The links of E' are sorted in decreasing order of weight. Starting from the maximum weight edge two adjacent clusters are merged sequentially. This is repeated as long as connected components are there in the graph and the number of clusters is greater than n. At the end of this step we get a set of clusters. This step takes $O(m \log m)$ time, where m is the number of edges and $m < n$. Some steps of this part are shown in Figure 4.

4. SAMPLE RESULTS

Figure 4 and Figure 4 provide a demonstration of our work. The figure on the top left shows an input query image. Perhaps we would like to obtain images from the Internet that has an action similar to the two baseball players.

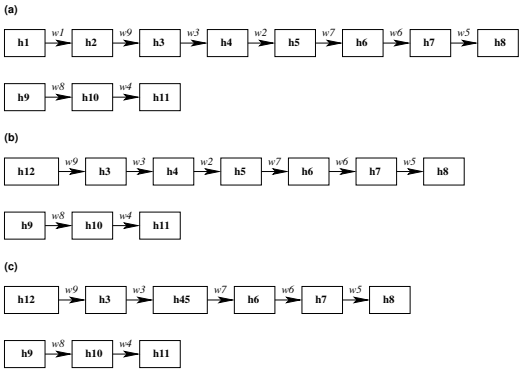

Figure 4. Illustration of Step 2. Handles are arranged as linked lists and edges are annotated according to their weights $w1 > w2 > w3 \ldots$. (a) Initial list after step 1, (b) List after merging $h1$ and $h2$ which are joined by maximum weight edge $w1$, (c) List after merging $h4$ and $h5$ which are joined by $w2$.

The middle figure (best seen in color) shows the result of Algorithm-P. It runs fast, but produces too many clusters. It would not be reasonable to give these clusters to an Internet search engine because it would increase the time to retrieve results, and possibly the quality too. Algorithm N-cut (top right, reproduced from the original paper) produces good results in identifying the figure. We were unable to produce this segmentation in a reasonable amount of time (our computer implementation ran out of memory also). The combined algorithm produces relevant portions as shown in the bottom.

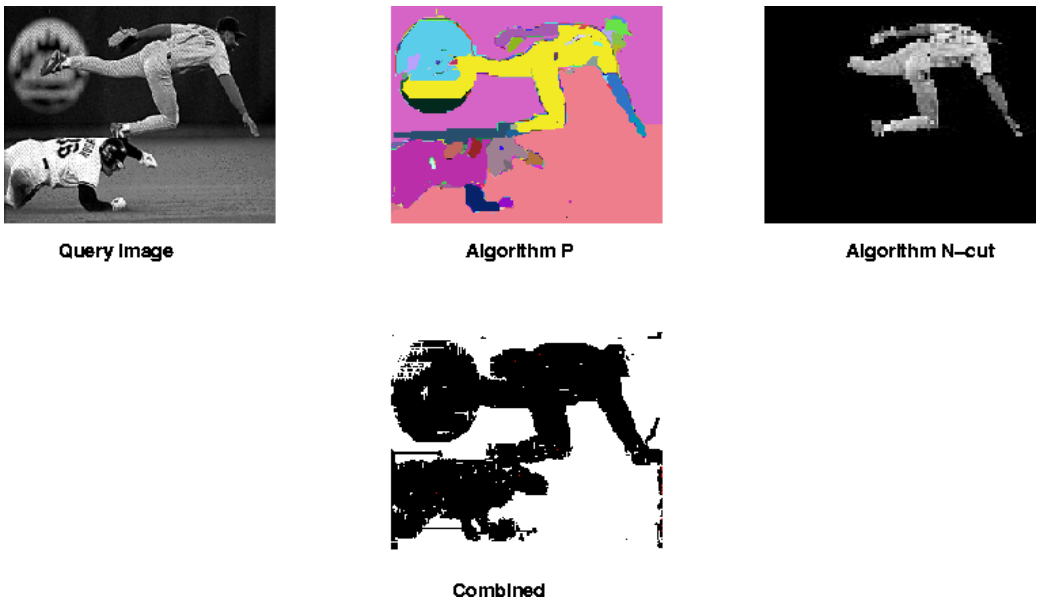

Figure 5. Region created using various algorithms. Figures are best seen in color.

A snapshot of our prototype system appears in Figure 4.

5. FINAL REMARKS

Image and video retrieval based on content from digital libraries, multimedia databases, the Internet, and other sources has been an important problem addressed by several researchers. In this regard, one cannot

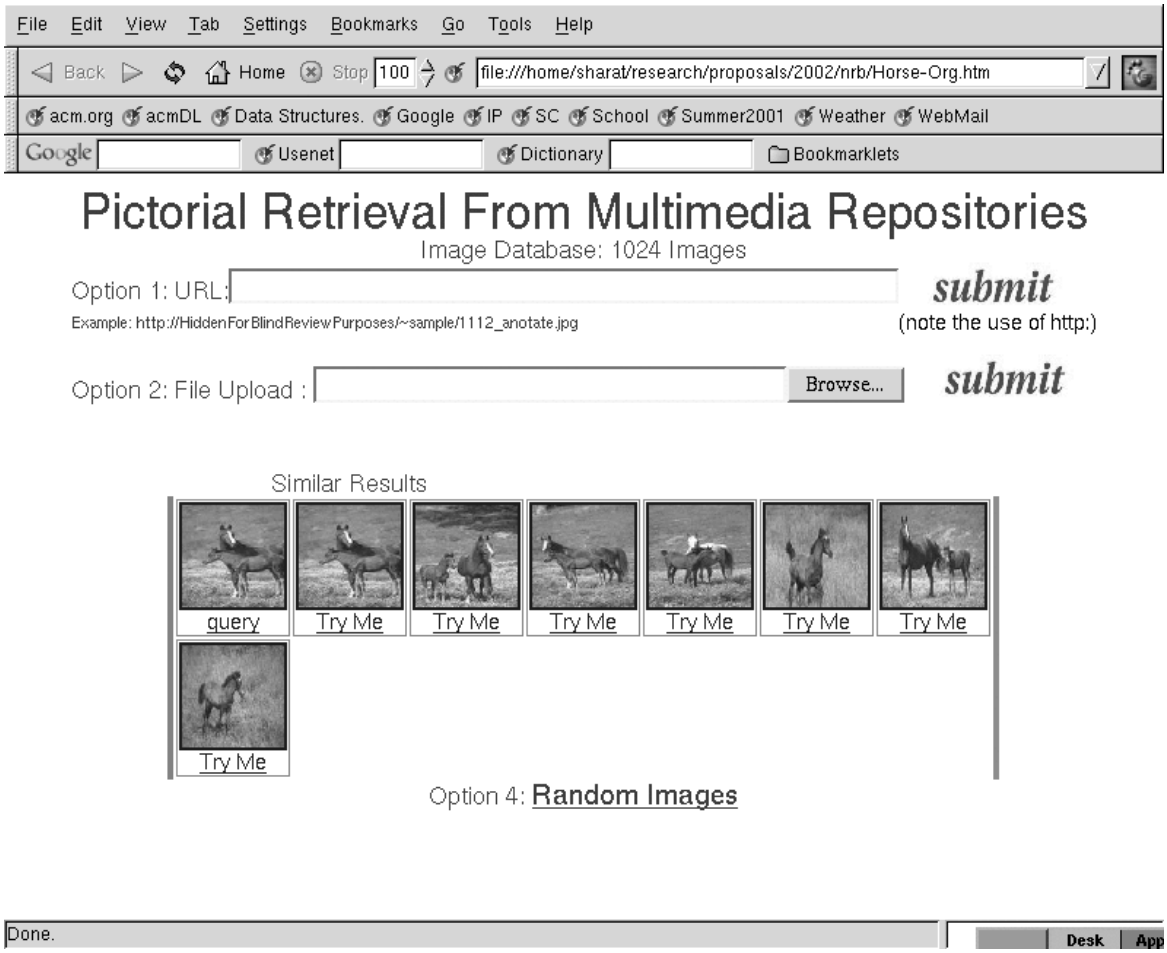

Figure 6. Prototype system available for Internet search.

overestimate the use of appropriate features such as color, texture, and shape. However, it has also become increasingly evident that the decomposition of images into regions is critical for useful results. In this paper we concentrate on producing regions from the point of view of Internet imaging. Once regions are produced, there are several algorithms and systems (including those produced in our research) that will output over the Internet images and video sequences that match user search terms (which are now sketch or image based).

Several region or segmentation algorithm have been proposed by researchers over the last three decades. The basic notion that we propose in this work is that, unlike the general segmentation problem, demands such as swift but approximate retrieval requires coarse controlled segmentation. That is, based on circumstances such as network congestion, and user profiles, we would like the automatic segmentation algorithm to make decisions on how coarse the segmentation should be made.

While there are several schools on how to produce regions, the graph theoretic approach has gained prominence in the recent literature. These approaches combine concepts such as normalized cut and spectral graph theory and provide more natural segmentation. Specifically, we get a quick idea of the foreground and background using this approach. This hierarchical approach is also useful for providing the hooks for controlling the level of segmentation. Unfortunately despite faster computers, and use of clever numerical techniques, these algorithms still remain beyond mainstream Internet usage.

A different local approach using spanning trees runs fast, but produces too many segments. It thus poses a burden to the matching routine, and worse, often results in too many false positives. It is not possible to control

the level of segmentation during the course of the algorithm.

In this work we combine the two approaches and retain the advantages of both approaches. The key idea is to use local approach for reducing the size of the problem that is fed to the normalized cut approach. Our algorithm runs in $O(n \log n)$ time. We believe this paradigm is useful for Internet imaging.

ACKNOWLEDGMENTS

We thank Biswarup Choudhury, Satwik Hebbar, Vishal Mamania, Abhineet Sawa, and Appu Shaji for their help in the implementation and discussions. Much of the implementation was based on original work by Naga Kiran.

REFERENCES

1. P. Felzenszwalb and D. Huttenlocher, "Image segmentation using local variation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 98–104, 1998.
2. J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8), pp. 888–905, 2000.
3. M. Pavan and M. Pelillo, "A new Graph-Theoretic approach to clustering and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 98–104, 2003.
4. J. Wills, S. Agrawal, and S. Belongie, "What went where," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 98–104, 2003.
5. Y. Rui, T. S. Huang, and S. Mehrotra, "Relevance feedback techniques in interactive content-based image retrieval," in *Storage and Retrieval for Image and Video Databases (SPIE)*, pp. 25–36, 1998.
6. M. Flickner, H. Sawhney, and W. Niblack, "Query by image and video content: The QBIC system," *IEEE Computer Magazine* **28**, pp. 23–32, 1995.
7. A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *International Journal of Computer Vision* **18**, pp. 233–254, 1996.
8. J. Wang, G. Wiederhold, O. Firschein, and S. Wei, "Wavelet-based image indexing techniques with partial sketch retrieval capability," in *Advances in Digital Libraries*, pp. 13–24, 1997.
9. J. Z. Wang, G. Wiederhold, and O. Firschien, "System for screening objectionable images using daubechies' wavelets and color histograms," *Computer Communications* , pp. 1355–1360, 1998.
10. C. Jacobs, A. Finkelstein, and D. Salesin, "Fast multiresolution image querying," *Computer Graphics* **29**, pp. 277–286, 1995.
11. M. Ramos, S. Hemami, and M. Tamburro, "Psychovisually-based multiresolution image segmentation," in *Proceedings of the IEEE International Conference on Image Proceedings*, pp. 66–69, 1997.
12. C. Carson, M. Thomas, S. Belongie, J. Hellerstein, and J. Malik, "Blobworld: a system for region-based image indexing and retrieval," in *Third International Conference on Visual Information Systems*, pp. 509–516, 1999.
13. J. Z. Wang and G. Wiederhold, "Simplicity: Semantics-sensitive Integrated Matching for Picture Libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 2001.
14. J. Li, J. Wang, and G. Wiederhold, "IRM: Integrated Region Matching for Image Retrieval," in *Proceedings of the 2000 ACM Multimedia Conference*, pp. 147–156, 2000.
15. S. Ardizzoni, I. Bartolini, and M. Patella, "Windsurf: Region-based image retrieval using wavelets," in *DEXA Workshop*, pp. 167–173, 1999.
16. J. Smith and S.-F. Chang, "Integrated spatial and feature image query," *ACM Multimedia* , 1996.
17. M. Das, Manmatha, and E. Riseman, "Indexing flower patent images using domain knowledge," *IEEE Intelligent Systems* , 1999.
18. K. Shearer, H. Bunke, and S. Venkatesh, "Video indexing and similarity retrieval by largest common subgraph detection using decision trees," *Pattern Recognition* , April 2000.
19. S. Aksoy and R. M. Haralick, "A graph-theoretic approach to image database retrieval," in *Visual Information and Information Systems*, pp. 341–348, 1999.
20. S. Chandran and K. K. Madheshia, "A fast segmentation algorithm revisited," *Proceedings of Indian Conference on Computer Vision, Graphics, and Image Processing* , December 2002.