

# A World Wide Web Region-Based Image Search Engine

Ioannis Kompatsiaris, Evangelia Triantafyllou and M. G. Strintzis

Centre for Research and Technology-Hellas/Informatics and Telematics Institute (CERTH/ITI)  
1st Km. Thermi-Panorama Road  
57001 Thermi-Thessaloniki, Greece.

Tel. : +30310996351, Fax : +30310996342,  
Email: strintzi@eng.auth.gr

## ABSTRACT

In this paper the development of an intelligent image content-based search engine for the World Wide Web is presented. Information Web Crawlers continuously traverse the Internet and collect images that are subsequently indexed based on integrated feature vectors. As a basis for the indexing, a novel K-Means segmentation algorithm is used, modified so as to take into account the coherence of individual regions. Based on the extracted regions, characteristic features are estimated using color, texture and shape/region boundary information. These features along with additional information are stored in a database. The user can access and search this indexed content through the Web with an advanced interface. Experimental results demonstrate the performance of the system, which can be reached in a publicly accessible web site.

## 1 INTRODUCTION

Very few tools are currently available for searching for images and videos over the Internet. This absence is particularly notable given the highly visual and graphical nature of the Web [1]. As with Web documents in general, the publication of visual information is highly volatile. In order to allow efficient search of the visual information, highly efficient automated systems are needed that regularly traverse the Web, detect visual information and process it in such a way to allow for efficient and effective search and retrieval [2].

Some popular text-based Internet search engines [3] offer restricted versions of such systems. In most of

them very simple indexing and retrieval algorithms are used for specific domains of a restricted set of images. In the commercial domain, IBM QBIC [4] is one of the earliest developed systems. Recently, additional systems have been developed at IBM T.J. Watson [5], VIRAGE [6], and NEC CC Research Labs [7]. Other research experiments [8, 9, 10] are far from being complete, lacking development in image collection and database management.

The paper is organized as follows. In the following section an overview of the proposed system architecture is presented. In Section 3 a description of the Information Crawlers is given, while the indexing and retrieval algorithms are presented in Section 4. Experimental results are presented in Section 5. Finally, conclusions are drawn in Section 6.

## 2 GENERAL SYSTEM ARCHITECTURE OVERVIEW

In this paper, ISTORAMA, a region-based color image indexing and retrieval system for the Internet, is presented. The overall system is split into two parts: (i) the off-line part and (ii) the on-line or user part.

In the off-line part, *Information Crawlers*, implemented entirely in Java [11], continuously traverse the WWW, collect images and transfer them to the central *Server* for further processing (Fig. 1). Then the *image indexing* algorithms process the image in order to extract descriptive features. Based on the extracted by the modified K-means algorithm [12, 13] regions, characteristic features are estimated using color, texture and

shape/region boundary information. The characteristic features along with information regarding the images such as the URL, date of transaction, size and a thumbnail are then stored in the database.

In the on-line part, a user connects to the system through a common Web Browser using the HTTP protocol. The user can then submit queries either by example images or by simple image information. The query is processed by the server and the *retrieval* phase begins; the indexing procedure is repeated again for the submitted image and then the extracted features are matched against those stored in the database using an SQL query. The results containing the URL as well as the thumbnail of the similar images are transmitted to the user by creating dynamic HTML pages. The results are ranked according to their similarity to the submitted image. In the following sections detailed descriptions of each component of the system will be presented.

### 3 INFORMATION CRAWLER

The image collection process is conducted by an autonomous Web agent or Crawler [14]. The agent traverses the Web by following the hyperlinks between documents. It detects images, retrieves and transfers them for processing to the system server. The extracted features are then added to the database. The overall collection process is carried out using several distinct modules:

- The Traversal Crawler - assembles lists of candidate Web pages that may include images or hyperlinks to them.
- The Hyperlink Parser - extracts the URLs of the images.
- The Retrieval Crawler - retrieves and transfers the image to the system server for further processing.

In the first phase, the Traversal Crawler traverses the Web looking for images. Starting from seed URLs, the Traversal Crawler follows a breadth-first search across the Web. It retrieves pages via Hypertext Transfer Protocol (HTTP) and passes the Hypertext Markup Language (HTML) code to the Hyperlink Parser. In turn, the Hyperlink Parser detects new URLs, encoded as HTML hyperlinks, and adds them back to the queue of Web pages to be retrieved by the Traversal Crawler.

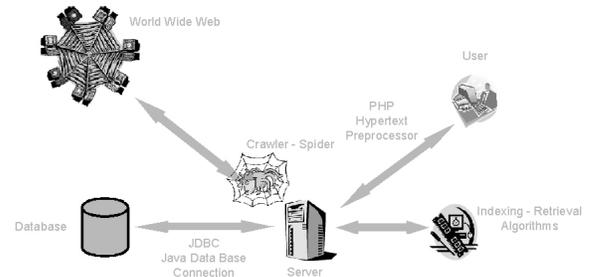


Figure 1: General System Architecture.

In this sense, the Traversal Crawler is similar to many of the conventional spiders or robots that follow hyperlinks in some fashion across the Web [15]. The Hyperlink Parser detects the hyperlinks in the Web documents and converts the relative URLs to absolute addresses. By examining the types of the hyperlinks and the filename extensions of the URLs, the Hyperlink Parser extracts the URLs of the images.

In the second phase, the list of image URLs from the Hyperlink Parser is passed to the Retrieval Crawler. The Retrieval Crawler retrieves the images and provides them as input to the indexing module. After the indexing procedure, the extracted features are added to the database. Another important function of the Retrieval Crawler is to extract attributes associated with the image such as URL, date of processing, size, width, height, and so forth, and also generate a thumbnail icon, that sufficiently compacts the visual information into a representative form.

## 4 IMAGE INDEXING AND RETRIEVAL

### 4.1 Region Extraction

After the image collection and transfer by the Information Crawler to the server, image indexing algorithms are used in order to extract descriptive features. Based on the extracted regions, characteristic features are extracted using color, texture and shape/region boundary information. As a basis for the indexing, a novel K-Means algorithm is used. Clustering based on the K-Means algorithm [16] tends to produce unconnected regions. This is due to the propensity of the classical K-Means algorithm to ignore spatial information about the intensity values in an image, since it only takes into account the global intensity or color information. In order to alleviate this problem, we propose the

use of an extended K-Means algorithm: the K-Means-with-connectivity-constraint algorithm (KMC). In this algorithm, both the *spatial proximity* of each region and the *texture* in the neighborhood of each pixel are also taken into account by defining two new centers for the K-Means algorithm and by integrating the K-Means with a component labeling procedure. In this paper a new color distance is defined for this algorithm. The  $L^*a^*b^*$  colour space is used [17], which is a suitable choice for indexing and retrieval applications because it is a perceptually equalized colour space, i.e. the numerical distance in this space is proportional to the perceived colour difference.

Texture properties in the neighborhood of each pixel are characterized using the Discrete Wavelet Frames (DWF) decomposition, proposed by [18]. A two-dimensional DWF decomposition of two levels of decomposition ( $L_d = 2$ ) that employs the low-pass Haar filter has been chosen [19]. Since three detail components are produced for each level of decomposition and each one of the three intensity components of the image, the texture feature vector for pixel  $\mathbf{p}$ ,  $T(\mathbf{p})$ , is composed of 18 texture components,  $\sigma_q(\mathbf{p})$ ,  $q = 1, \dots, 18$ :

$$T(\mathbf{p}) = [\sigma_1(\mathbf{p}), \sigma_2(\mathbf{p}), \dots, \sigma_{18}(\mathbf{p})]^T \quad (1)$$

The K-Means with connectivity constraint (KMC) algorithm consists of the following steps:

- Step 1 The classical KM algorithm is performed for a small number of iterations. This result in  $K$  regions, with color centers  $\bar{\mathbf{I}}_k$  defined as:

$$\bar{\mathbf{I}}_k = \frac{1}{M_k} \sum_{m=1}^{M_k} \mathbf{I}(\mathbf{p}_m^k), \quad (2)$$

where  $\mathbf{I}(\mathbf{p})$  are the color components of pixel  $\mathbf{p}$  in the  $L^*a^*b^*$  color space, i.e.  $\mathbf{I}(\mathbf{p}) = (I_L(\mathbf{p}), I_a(\mathbf{p}), I_b(\mathbf{p}))$ . Spatial centers  $\bar{\mathbf{S}}_k = (\bar{S}_{k,X}, \bar{S}_{k,Y})$ ,  $k = 1, \dots, K$  and texture centers  $\bar{\mathbf{T}}_k = (\sigma_1(\mathbf{p}), \sigma_2(\mathbf{p}), \dots, \sigma_{18}(\mathbf{p}))$  for each region are defined as follows:

$$\bar{S}_{k,X,Y} = \frac{1}{M_k} \sum_{m=1}^{M_k} p_{m,X,Y}^k, \quad (3)$$

$$\bar{\mathbf{T}}_k = \frac{1}{M_k} \sum_{m=1}^{M_k} T(\mathbf{p}_m^k), \quad (4)$$

where  $\mathbf{p}^k = (p_X^k, p_Y^k)$ . The area of each region  $A_k$  is defined as  $A_k = M_k$  and the mean area of all regions  $\bar{A} = \frac{1}{K} \sum_{k=1}^K A_k$ .

- Step 2 For every pixel  $\mathbf{p} = (x, y)$  the color differences are evaluated between center and pixel colors as well as the distances between  $\mathbf{p}$  and  $\bar{\mathbf{S}}$ . A generalized distance of a pixel  $\mathbf{p}$  from a subobject  $s_k$  is defined as follows:

$$D(\mathbf{p}, k) = \|I(\mathbf{p}) - I_k\| + \|T(\mathbf{p}) - T_k\| + \lambda \frac{\bar{A}}{A_k} \|\mathbf{p} - S(s_k)\|, \quad (5)$$

where  $\|\mathbf{p} - \bar{\mathbf{S}}_k\|$  is the Euclidean distance,  $\sigma_I^2, \sigma_S^2$  are the standard deviations of color and spatial distance, respectively and  $\lambda$  is a regularization parameter, which defined as:

$$\lambda = 0.4 \cdot \frac{Db_{max}}{\sqrt{p_{x,max}^2 + p_{y,max}^2}}$$

Normalization of the spatial distance,  $\|\mathbf{p} - \bar{\mathbf{S}}_k\|$  by dividing by the area of each subobject,  $\frac{\bar{A}}{A_k}$  is necessary in order to allow the creation of large connected objects; otherwise, pixels with similar color and motion values with those of large object would be assigned to neighboring smaller regions. If  $|D(\mathbf{p}, i)| < |D(\mathbf{p}, k)|$  for all  $k \neq i$ ,  $\mathbf{p} = (x, y)$  is assigned to region  $s_i$ .

- Step 3 Based on the above subdivision, an eight connectivity component labeling algorithm is applied. This algorithm finds all connected components and assigns a unique value to all pixels in the same component. Regions whose area remains below a predefined threshold are not labeled as separate regions. The component labeling algorithm produces  $L$  connected regions. For these connected regions, the color  $\bar{\mathbf{I}}_l$  and spatial  $\bar{\mathbf{S}}_l$  and motion centers  $l = 1, \dots, L$ , are calculated using equations (2) and (3) respectively.
- Step 4 If the difference between the new and the old centers  $\bar{\mathbf{I}}_l$  and  $\bar{\mathbf{S}}_l$  is below a threshold, then stop, else goto Step 2 with  $K = L$  using the new color and spatial centers.

Through the use of this algorithm the ambiguity in the selection of number  $K$  of regions, which is another shortcoming of the K-Means algorithm, is also resolved. Starting from any  $K$ , the component labeling algorithm produces or rejects regions according to their compactness. In this way  $K$  automatically adjusted during the segmentation procedure.

## 4.2 Region Descriptors

We store a simple description of each region's color, texture and spatial characteristics.

In order to represent the color distribution of each region, we store the color histogram of the pixels in the region. This histogram is based on bins with width 20 in each dimension of  $L^*a^*b^*$  space. This spacing yields five bins in the  $L^*$  dimension and ten bins in each of the  $a^*$  and  $b^*$  dimensions. For each image region we also store the mean texture descriptors (i.e., anisotropy, orientation, contrast).

The geometric descriptors of the region are simply the spatial center  $\bar{\mathbf{S}}_k$  and covariance or scatter matrix  $\mathbf{C}_k$  of the region. The centroid  $\bar{\mathbf{S}}_k$  provides a notion of position, while the scatter matrix provides an elementary shape description. In the querying process discussed in Section 4.3, centroid separations are expressed using Euclidean distance. The determination of the distance between scatter matrices, which is slightly more complicated, is based on the three quantities  $[\det(S)]^{1/2} = \sqrt{\rho_1\rho_2}$ ,  $1 - \rho_1/\rho_2$  and  $\theta$  where  $\rho_1$  and  $\rho_2$  are the eigenvalues and  $\theta$  the argument of the principal eigenvector of  $\mathbf{C}_k$ . These three quantities represent approximate area, eccentricity and orientation.

Specifically, if  $\mathbf{p}_m^k = [p_{m,X}^k, p_{m,Y}^k]^T$ ,  $m = 1, \dots, M_k$  are the pixels belonging to region  $k$  with coordinates  $p_{m,X}^k, p_{m,Y}^k$  then the covariance (or scatter) matrix of region  $k$  is

$$\mathbf{C}_k = \frac{1}{M_k} \sum_{m=1}^{M_k} (\mathbf{p}_m^k - \bar{\mathbf{S}}_k) (\mathbf{p}_m^k - \bar{\mathbf{S}}_k)^T.$$

Let  $\rho_i, \mathbf{u}_i, i = 1, 2$  be its eigenvalues and eigenvectors:  $\mathbf{C}_k \mathbf{u}_i = \rho_i \mathbf{u}_i$  with  $\mathbf{u}_i^T \mathbf{u}_i = 1$ ,  $\mathbf{u}_i^T \mathbf{u}_j = 0$ ,  $i \neq j$  and  $\rho_1 \geq \rho_2$ . As is known from Principal Component Analysis (PCA), the principal eigenvector  $\mathbf{u}_1$  defines the orientation of the region and  $\mathbf{u}_2$  is perpendicular to  $\mathbf{u}_1$ . The two eigenvalues provide an approximate measure of the two dominant directions of the shape.

## 4.3 Image Retrieval by Querying

In our system the user composes a query by submitting an image to the segmentation/feature extraction algorithm in order to see its segmented representation, selecting the regions to match, and finally specifying the relative importance of the region features. Once a query is specified, we score each database image on the basis of how closely it satisfies the query. The score  $\mu_i$  for each query is calculated as follows:

1. Find the feature vector  $f_i$  for the desired region  $s_i$ . This vector consists of the stored color, position, and shape descriptors (Section 4.2).
2. For each region  $s_j$  in the database image:
  - (a) Find the feature vector  $f_j$  for  $s_j$ .
  - (b) Find the Mahalanobis distance between  $f_i$  and  $f_j$  using the diagonal covariance matrix (feature weights)  $\Sigma$  defined by the user:  $d_{ij} = [(f_i - f_j)^T \Sigma^{-1} (f_i - f_j)]^{1/2}$ .
  - (c) Measure the similarity between  $f_i$  and  $f_j$  using the index  $\mu_{ij} = e^{-\frac{d_{ij}}{2}}$ . This index is 1 if the regions are identical in all relevant features; it decreases as the match becomes less perfect.
3. Calculate  $\mu_i = \max_j \mu_{ij}$ .

The images are then ranked according to the overall score and the best matches are returned, along with their related information such as URL, date, size, etc.

## 5 EXPERIMENTAL RESULTS

The proposed system was used for image collection, segmentation into regions and region feature extraction for images from real Web sites and for test images. The system is accessible through WWW at the following address: <http://uranus.ee.auth.gr/Istorama>. An important part of the system is the segmentation algorithm. The performance of this part greatly affects the performance of the entire system. As can be seen from Fig. 2, which shows the final segmentation of 8 randomly selected images, the segmentation results are good for natural images appearing on the Web, since the distinct objects are separated from the background while over-segmentation does not occur. At the same time the details and sharpness of important edges are retained. In

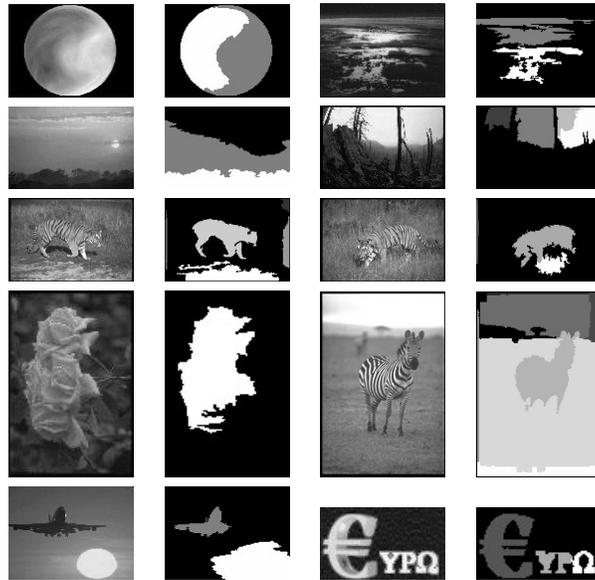


Figure 2: Images segmented into regions.

case of over-segmentation, the system treats the whole image as one region.

The user accesses the system through the home page and in order to select the query image, he can search in the database based on a keyword, browse the existing categories, or upload an image.

After the selection of the query image, the user has to select a region in the image that is of interest. Unlike with text search engines, in which the user can see the features (words) in a document, very few of the current image retrieval systems allow the user to see exactly what the system is looking for in response to a query. But if the user is not aware of the manner in which input image was improperly processed, the user can only guess as to what went wrong. In order to help the user formulate effective queries and understand their results, as well as to minimize disappointment due to overly optimistic expectations of the system, our system displays its segmented representation of the submitted image and allows the user to specify which aspects of that representation are relevant to the query. Therefore, when the desired region has been selected, the user is allowed to adjust the weights of each feature of the selected region.

In order to evaluate the performance of the system, we performed a variety of queries using a set of 3,500 images (some of which were from the commercial Corel stock photo collection and the others were images from real websites). Sample queries are shown in Fig. 3. As can be seen from the results, the returned images match the selection criteria of the user. Concerning computational efficiency of the system, on a Pentium II 450MHz PC using the Linux operating system, four seconds are needed on the average to segment a  $128 \times 96$  image and to compute the features of all regions. The matching procedure is also efficient. When the query image is in the database, it takes about one second on the average to retrieve a set of similar images from the 3,500-image database using our similarity measure. If the query image is not in the database, four extra seconds are spent to process the query (segment and extract features).

### 5.1 Comparison to global histogram

The color histogram is often used in image retrieval systems due to its good performance in characterizing the global color content [20]. It is easy to compute with only few constraints when applied to images. However,

istorama CONTENT BASED IMAGE RETRIEVAL OVER THE INTERNET Home | About | Contact

## Image Search

Content based search for: **Results**



Found 5 images ...

 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 100% Filename: 777 Size: 128x85, 2.3Kb	 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 94% Filename: 788 Size: 85x129, 2.5Kb	 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 92% Filename: 1008 Size: 128x85, 1.6Kb
 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 92% Filename: 1795 Size: 128x85, 1.6Kb	 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 88% Filename: 1292 Size: 128x96, 1.9Kb	

Query for the region of the sun in sunset.

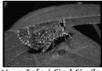
istorama CONTENT BASED IMAGE RETRIEVAL OVER THE INTERNET Home | About | Contact

## Image Search

Content based search for: **Results**



Found 126 images [Images #1 to #9]... Previous Page | Next Page

 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 100% Filename: 1630 Size: 128x85, 3.7Kb	 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 96% Filename: 1624 Size: 128x85, 4.0Kb	 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 95% Filename: 1606 Size: 128x85, 4.1Kb
 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 95% Filename: 77 Size: 128x85, 3.3Kb	 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 95% Filename: 1596 Size: 128x85, 3.9Kb	 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 94% Filename: 1609 Size: 128x85, 3.5Kb

Query for a car.

istorama CONTENT BASED IMAGE RETRIEVAL OVER THE INTERNET Home | About | Contact

## Image Search

Content based search for: **Results**



Found 3 images ...

 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 100% Filename: pcrmag_0ma00 Size: 96x180, 5.5Kb	 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 95% Filename: pworldclass Size: 150x51, 2.0Kb	 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 87% Filename: itv Size: 150x76, 2.3Kb
--	--	--

Query for the letter P.  
The system finds this letter in different words.

istorama CONTENT BASED IMAGE RETRIEVAL OVER THE INTERNET Home | About | Contact

## Image Search

Content based search for: **Results**



Found 59 images [Images #1 to #9]... Previous Page | Next Page

 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 100% Filename: 2350 Size: 128x128, 4.1Kb	 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 96% Filename: 2059 Size: 128x112, 4.3Kb	 <a href="#">More Info</a>   <a href="#">Find Similar</a> Similarity: 94% Filename: 2064 Size: 128x112, 4.2Kb
--	--	--

Query for the earth.  
The system retrieves the required pattern.

Figure 3: Query Results.

it has several inherent problems in image indexing and retrieval. Most importantly, it does not include any spatial information.

We expected that querying in our system would perform better than histogram comparison. In order to test this conjecture, we selected seven object categories: People, Landscapes, Flowers, Butterflies, Sea, Web images, and Space. There were between 30 and 100 examples of each category among the 3,500 images.

For color histogram match we used the same color space ( $L^*a^*b^*$ ) and the same 25 bins along with the same distance. For each category we assessed a set of 18 queries (assessment achieved by 3 different persons), considered as the ground truth against which we evaluated our system.

For each category we calculated the average recall and precision, according to the results of the 18 queries. Recall is defined as the number of retrieved relevant images over the total number of relevant images in the database. Precision is defined as the number of relevant images over the total number of retrieved images. In Fig. 4 the average precision is plotted vs. recall for the web images. (The 3,500 database images were categorized as “relevant” or “not relevant” to each category—i.e., containing or not the object, by a human observer.) The results supported our conjecture that our system would yield better results than histogram matching. For a system that ranks images randomly, the average precision is about 0.1. The global histogram matching performs much worse than that of our system, with average lower precisions in all image categories.

## 6 CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

In this paper the development of an intelligent image content-based search engine for the World Wide Web was presented. Information Web Crawlers continuously traverse the Internet and collect images that are subsequently indexed based on integrated feature vectors. As a basis for the indexing, the K-Means algorithm is used, modified so as to take into account the coherence of the regions. Based on the extracted regions, characteristic features are estimated using color, texture and shape/region boundary information. These features along with additional information are stored in a database. The user can access and search this indexed

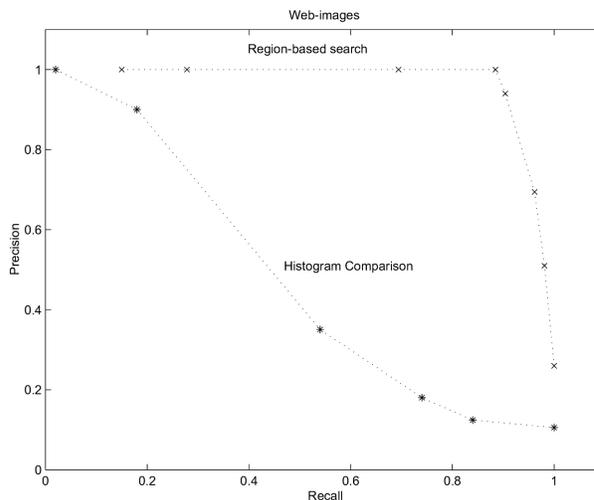


Figure 4: Precision vs. Recall curve for the categories “web images”.

content through the Web. The output of the system is a set of links to the content available in the WWW, ranked according to their similarity to the image submitted by the user.

The proposed application can be used in existing text-based WWW search engines in order to assist users to search for visual information. It can be also used in order to track illegal usage of multimedia content over the Internet; a multimedia content owner can search and find web pages where the content appears and determine illegal usage. The system could be easily integrated with watermarking detection algorithms in order to automatically track unauthorized usage.

In future work, we will emphasize on other visual dimensions. We also plan to extend our technique so as to handle video indexing by extracting motion features. The standardized format of MPEG-7 for the description of the features will also be used and stored in the database.

## 7 ACKNOWLEDGEMENTS

This work was supported by the Greek Secretariat for Research and Technology projects PENED99 and PABE . The assistance of COST 211 quat and the IST project Network of Excellence SCHEMA (IST-2001-32795) is also gratefully acknowledged. We fi-

nally wish to thank INTRASOFT North Greece S.A. (<http://www.intrasoft.gr/>) for their invaluable assistance and Lambros A. Makris for his contribution to the implementation of the online query system and for many useful discussions.

#### REFERENCES

- [1] J. R. Smith and S.-F. Chang, "Visually Searching the Web for Content," *IEEE Multimedia Magazine*, vol. 4, no. 3, pp. 12–20, Summer 1997.
- [2] B. Carlson, "Web still & video image content search & retrieval," *Advanced Imaging*, vol. 60, pp. 34–36, 1997.
- [3] L. Introna and H. Nissenbaum, "Defining the Web: the politics of search engines," *IEEE Computer*, vol. 33, no. 1, pp. 54–62, January 2000.
- [4] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "The QBIC project: querying images by content using color, texture, and shape," in *Proc. SPIE*, San Jose, February 1993, vol. 1908, pp. 173–87.
- [5] J. R. Smith and C. S. Li, "Image classification and querying using composite region templates," *Journal of Computer Vision and Image Understanding*, vol. 75, no. 1–2, pp. 165–174, 1999, Academic press.
- [6] A. Gupta and R. Jain, "Visual Information retrieval," *Communications of the ACM*, vol. 40, no. 5, pp. 70–79, May 1997.
- [7] S. Mukherjea, K. Hirata, and Y. Hara, "AMORE: a World Wide Web image retrieval engine," *World Wide Web*, vol. 2, no. 3, pp. 115–32, 1999, Baltzer.
- [8] J. R. Smith and S.-F. Chang, "An image and video search engine for the World-Wide Web," in *Proc. SPIE*, 1997, vol. 3022, pp. 84–95.
- [9] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," in *Proc. Int. Conf. on Visual Information Systems*, Amsterdam, The Netherlands, June 1999.
- [10] J. S. Wang, G. Wiederhold, O. Firschein, and X. W. Sha, "Content-based image indexing and searching using Daubechies' wavelets," *International Journal of Digital Libraries*, vol. 1, no. 4, pp. 311–328, 1998.
- [11] K. Arnold and J. Gosling, *The Java Programming Language*, Reading. Addison-Wesley, Mass., 2nd edition, 1997.
- [12] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal Segmentation and Tracking of Objects for Visualization of Videoconference Image Sequences," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 8, December 2000.
- [13] I. Kompatsiaris, E. Triantafillou, and M. G. Strintzis, "Region-Based Color Image Indexing and Retrieval," in *2001 International Conference on Image Processing (ICIP2001)*, Thessaloniki, Greece, October 7-10 2001.
- [14] T. M. Ciolek, "Today's WWW-tomorrow's MMM? The specter of multi-media mediocrity," *IEEE Computer*, vol. 29, no. 1, pp. 106–108, January 1996.
- [15] M. Koster, "Robots in the web: threat or treat?," *Connections*, vol. 9, no. 4, April 1995.
- [16] I. Kompatsiaris, G. Mantzaras, and M. G. Strintzis, "Spatiotemporal Segmentation and Tracking of Objects in Color Image Sequences," in *2000 IEEE International Symposium on Circuits and Systems (ISCAS 2000)*, Geneva, Switzerland, May 2000.
- [17] S. Liapis, E. Sifakis, and G. Tziritas, "Color and/or Texture Segmentation using Deterministic Relaxation and Fast Marching Algorithms," in *Intern. Conf. on Pattern Recognition*, September 2000, vol. 3, pp. 621–624.
- [18] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. on Image Processing*, vol. 4, no. 11, pp. 1549–1560, November 1995.
- [19] N. V. Boulgouris, I. Kompatsiaris, V. Mezaris, D. Simitopoulos, and M. G. Strintzis, "Segmentation and Content-based Watermarking for Color Image and Image Region Indexing and Retrieval," *EURASIP Journal on Applied Signal Processing*, to appear, April 2002.
- [20] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.