

Region-Based Color Image Indexing and Retrieval

Kompatsiaris, Ioannis; Triantafyllou, Evangelia; Strintzis, Michael G.

Published in:

Proceedings of the IEEE International Conference on Image Processing 2001

Publication date:
2001

Document Version
Accepted author manuscript, peer reviewed version

[Link to publication from Aalborg University](#)

Citation for published version (APA):

Kompatsiaris, I., Triantafyllou, E., & Strintzis, M. G. (2001). Region-Based Color Image Indexing and Retrieval. In *Proceedings of the IEEE International Conference on Image Processing 2001*

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

REGION-BASED COLOR IMAGE INDEXING AND RETRIEVAL

Ioannis Kompatsiaris, Evagelia Triantafyllou and Michael G. Strintzis

Information Processing Laboratory
Electrical and Computer Engineering Department
Aristotle University of Thessaloniki
540 06 Thessaloniki, Greece

Informatics and Telematics Institute
1, Kyvernidou str.
546 39 Thessaloniki, Greece
Email: ikom@iti.gr

ABSTRACT

In this paper a region-based color image indexing and retrieval algorithm is presented. As a basis for the indexing, a novel K-Means segmentation algorithm is used, modified so as to take into account the coherence of the regions. A new color distance is also defined for this algorithm. Based on the extracted regions, characteristic features are estimated using color, texture and shape information. An important and unique aspect of the algorithm is that, in the context of similarity-based querying, the user is allowed to view the internal representation of the submitted image and the query results. Experimental results demonstrate the performance of the algorithm. The development of an intelligent image content-based search engine for the World Wide Web is also presented, as a direct application of the presented algorithm.

1. INTRODUCTION

Very large collections of images are growing ever more common. From stock photo collections to proprietary databases to the Web, these collections are diverse and often poorly indexed; unfortunately, image retrieval systems have not kept pace with the collections they are searching. Following recent developments, new multimedia standards, such as MPEG-4 and MPEG-7, do not concentrate only on efficient compression methods but also on providing better ways to represent, integrate and exchange visual information [1, 2]. These efforts aim to provide the user with greater flexibility for "content-based" access and manipulation of multimedia data.

The shortcomings of indexing and retrieval systems [3, 4, 5] are due both to the image representations they use and to their methods of accessing those representations to find images:

- While users often would like to find images containing particular objects, most existing image retrieval systems represent images based only on their low-level features, with little regard for the spatial organization of those features.
- Systems based on user querying are often unintuitive and offer little help in understanding why certain images were returned and how to refine the query. Often the user knows only that she has submitted a query for, say, a bear and retrieved very few pictures of bears in return.

This work was supported by the Greek Secretariat for Research and Technology projects PENED99 and PABE. The assistance of COST 211 quat is also gratefully acknowledged.

- For general image collections, there are currently no systems that automatically classify images or recognize the objects they contain.

In order to overcome these problems, a region-based color indexing and retrieval algorithm is presented [6, 7]. As a basis for the indexing, a novel K-Means segmentation algorithm is used, modified so as to take into account the coherence of the regions. A new color distance is also defined for this algorithm. The $L^*a^*b^*$ colour space is used [8], which is related to the CIE 1931 XYZ standard observer through a nonlinear transformation. The $L^*a^*b^*$ is a suitable choice because it is a perceptually equalised colour space, i.e. the numerical distance in this space is proportional to the perceived colour difference. Based on the extracted regions, characteristic features are estimated using color, texture and shape information. An important and unique aspect of the algorithm is that, in the context of similarity-based querying, the user is allowed to view the internal representation of the submitted image and the query results. More specifically, in a querying task, the user can access the regions directly in order to see the segmentation of the query image and specify which aspects of the image are central to the query.

The development of an intelligent image content-based search engine for the World Wide Web is also presented, as a direct application of the presented algorithm. Information Web Crawlers continuously traverse the Internet and collect images that are subsequently indexed based on the integrated feature vectors. These features along with additional information such as the URL location and the date of index procedure are stored in a database. The user can access and search this indexed content through the Web with an advanced and user friendly interface. 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 paper is organised as follows. In Section 2 the K-Means region extraction algorithm is presented. In Section 3 the descriptors of each region are given, while the querying procedure is presented in Section 4. Experimental results and the content-based search engine for the World Wide Web application are presented in Section 5.

2. REGION EXTRACTION

As a basis for the indexing, the K-Means algorithm is used. Clustering based on the K-Means algorithm is a widely used region segmentation method [9, 10] which, however 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. In this algorithm the *spatial proximity* of each region is also taken into account by defining a new center for the K-Means algorithm and by integrating the K-Means with a component labeling procedure.

For the sake of easy reference we shall first describe the traditional K-Means algorithm (KM):

- Step 1 For every region $s_k, k = 1, \dots, K$, random initial intensity values are chosen for the region intensity centers \bar{I}_k .
- Step 2 For every pixel $\mathbf{p} = (x, y)$, the difference is evaluated between $I(x, y)$, where $I(x, y)$ is the intensity value of the pixel and $\bar{I}_k, k = 1, \dots, K$. If $|I(x, y) - \bar{I}_i| < |I(x, y) - \bar{I}_k|$ for all $k \neq i$, $\mathbf{p}(x, y)$ is assigned to region s_i .
- Step 3 Following the new subdivision, \bar{I}_k is recalculated. If M_k elements are assigned to s_k then: $\bar{I}_k = \frac{1}{M_k} \sum_{m=1}^{M_k} I(\mathbf{p}_m^k)$, where $\mathbf{p}_m^k, m = 1, \dots, M_k$, are the pixels belonging to region s_k .
- Step 4 If the new \bar{I}_k are equal with the old then stop, else goto Step 2.

The results of the application of the above algorithm are improved using the K-Means with connectivity constraint (KMC) algorithm, which 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:

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

where $\mathbf{I}(\mathbf{p})$ are the color components of pixel \mathbf{p} in the Lab color space, i.e. $\mathbf{I}(\mathbf{p}) = (I_L(\mathbf{p}), I_a(\mathbf{p}), I_b(\mathbf{p}))$. Spatial centers $\bar{\mathbf{S}}_k = (\bar{S}_{k,X}, \bar{S}_{k,Y}), k = 1, \dots, K$ for each region are defined as follows:

$$\bar{S}_{k,X,Y} = \frac{1}{M_k} \sum_{m=1}^{M_k} p_{m,X,Y}^k, \quad (2)$$

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:

$$D(\mathbf{p}, k) = \frac{\lambda_1}{\sigma_I^2} \|\mathbf{I}(\mathbf{p}) - \bar{\mathbf{I}}_k\| + \frac{\lambda_2}{\sigma_S^2} \bar{A} \frac{\|\mathbf{p} - \bar{\mathbf{S}}_k\|}{A_k}, \quad (3)$$

where $\|\cdot\|$ is the Euclidean distance, σ_I^2, σ_S^2 are the standard deviations of color and spatial distance, respectively and λ_1, λ_2 are regularization parameters. Normalization of the spatial distance, $\|\mathbf{p} - \bar{\mathbf{S}}_k\|$ with the area of each subobject, $\frac{\bar{A}}{A_k}$ is necessary in order to allow the creation of large



Fig. 1. (a) Original image “Claire”, (b) Result of the KMC algorithm (Step 2, first iteration). (c) Result of the component labeling algorithm (Step 3). (d) The final segmentation after only four iterations.

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 (1) and (2) 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.

In Fig. 1 an example of the segmentation procedure is shown. In Fig. 1a the original image of the videoconference image “Claire” of size 176×144 is shown. In Fig. 1b the result of the first iteration of the KMC algorithm is shown (result of Step 2 of the KMC algorithm, for the first iteration). In Fig. 1c the result of the component labeling algorithm is shown (Step 3 of the algorithm). The initial number of subobject was set to $K = 5$ and the component labeling algorithm produced $L = 6$ regions. In Fig. 1d the final segmentation after only four iterations is shown.

3. REGION DESCRIPTORS

We store a simple description of each region’s color, texture and spatial characteristics. For each extracted region k , the color centers $\bar{\mathbf{I}}_k$ as estimated during the segmentation procedure are stored. 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, 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 θ (ρ_1

and ρ_2 are the eigenvalues and θ 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. IMAGE RETRIEVAL BY QUERYING

In our system, similarly to that in [11], 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 based on how closely it satisfies the query. The score μ_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 3).
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) Σ set 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 $\mu_{ij} = e^{-\frac{d_{ij}^2}{2}}$. This score is 1 if the regions are identical in all relevant features; it decreases as the match becomes less perfect.
3. Take $\mu_i = \max_j \mu_{ij}$.

We then rank the images according to the overall score and return the best matches, along with their related information.

5. EXPERIMENTAL RESULTS

The proposed algorithm has been used for indexing and retrieval of color images. In Fig. 3 the results of a query are shown. In Fig. 3a,b the input image and its segmented representation are shown, respectively. The segmented regions correspond to semantic objects, allowing efficient indexing and retrieval. Black areas of the segmented image correspond to unsorted regions. The user starts a query by selecting the facial region. The algorithm returns a set of images (the first image is always the submitted one) also containing the selected facial region and a fireworks image, which contains a region very similar in color and shape to the submitted one. Most of the results appear to have high semantic relevance with the submitted image. In Fig. 4 the user starts a query by

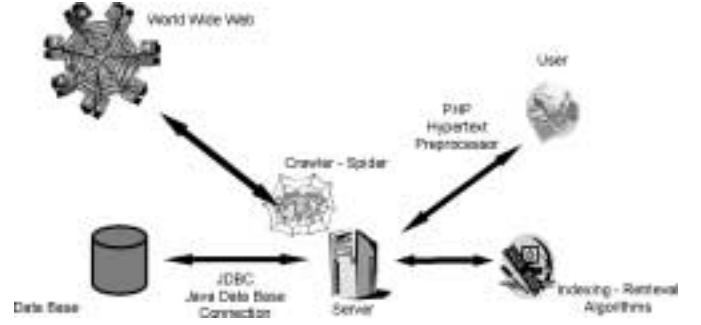


Fig. 2. General System Architecture.

selecting the sea region. The retrieved images contain a similar region, which is also a sea or a sky region.

The proposed algorithm can be used for the development of an intelligent image content-based search engine for the World Wide Web¹. To allow efficient search of the visual information, a highly automated system is needed that regularly traverses the Web, detects visual information and processes it in such away to allow for efficient and effective search and retrieval [5]. 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*, continuously traverse the WWW, collect images and transfer them to the central *Server* for further processing (Fig. 2). Then the *image indexing* algorithms process the image in order to extract descriptive features. These features along with information of the images such as URL, date of process, size and a thumbnail are stored in the *database*. In this stage the full size initial image is discarded.

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 (size, date, initial location, etc). The query is then 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. The results containing the URL as well as the thumbnail of the similar images are presented to the user by creating an HTML page using the PHP scripting language. The results are ranked according to their similarity to the submitted image.

6. REFERENCES

- [1] L. Chiariglione, "MPEG and Multimedia Communications," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 5–18, Feb. 1997.
- [2] R. Koenen and F. Pereira, "MPEG-7: A standardised description of audiovisual content," *Signal Processing: Image Communication*, vol. 16, no. 1–2, pp. 5–13, 2000.
- [3] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," in *SPIE Storage and Retrieval of Image & Video Databases II*, February 1994.
- [4] P.M. Kelly, T.M. Cannon, and D.R. Hush, "Query by image example: the CANDID approach," in *SPIE Storage and Retrieval for Image and Video Databases III*, 1995, vol. 2420, pp. 238–248.

¹For more information: <http://uranus.ee.auth.gr/Istorama>.

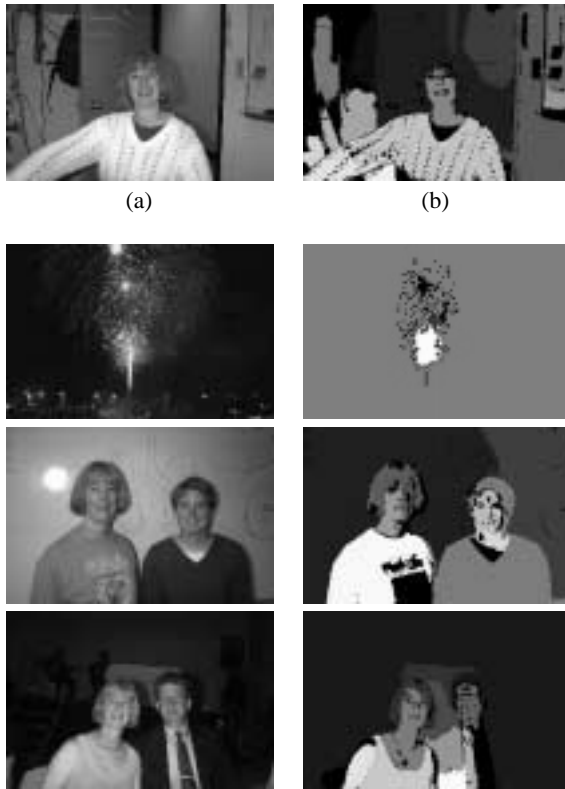


Fig. 3. Querying example: (a) Input image (b) Input image segmented into regions. The face region is selected. Result images along with their segmented representation. The results are sorted according to their similarity to the submitted image.

- [5] 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.
- [6] "Special Issue," *IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Image and Video Processing*, vol. 8, no. 5, Sept. 1998.
- [7] P. Salembier and F. Marques, "Region-Based Representations of Image and Video: Segmentation Tools for Multimedia Services," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1147–1169, December 1999.
- [8] 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.
- [9] S. Z. Selim and M. A. Ismail, "K-means-type algorithms," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 6, no. 1, pp. 81–87, Jan. 1984.
- [10] 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.
- [11] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Region-based image querying," in *CVPR '97 Workshop on Content-Based Access of Image and Video Libraries*, 1997.

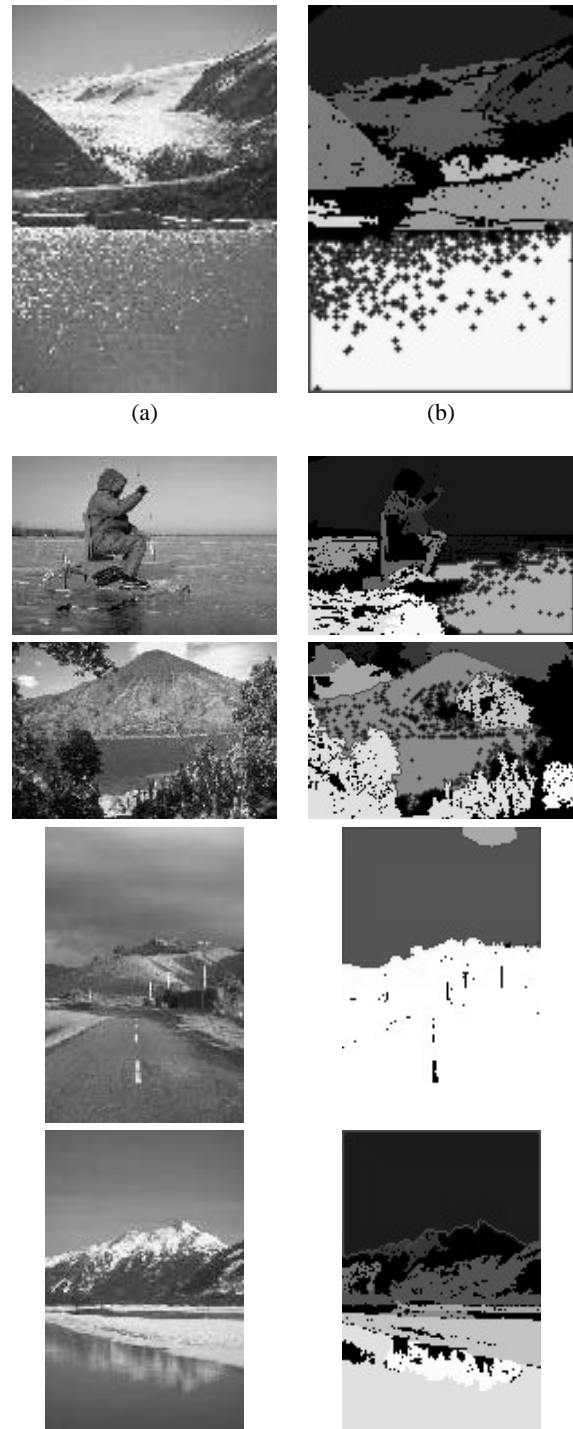


Fig. 4. Querying example: (a) Input image (b) Input image segmented into regions. The sea region is selected. Result images along with their segmented representation. The results are sorted according to their similarity to the submitted image.