A framework for the retrieval of multiple regions using Binary Partition Trees and low level descriptors

Luis Garrido and Philippe Salembier
Universitat Politècnica de Catalunya, Barcelona, SPAIN
oster@gps.tsc.upc.es, philippe@gps.tsc.upc.es

ABSTRACT

This paper proposes a framework for the retrieval of multiple regions characterized by low-level features. The retrieval combines the assessment of the visual similarity between regions and of the similarity of the relationship between these regions. Binary Partition Trees (BPTs) are used as a basis of the image representation. Regions of the BPT are described by low-level descriptors. Finally, relevance feedback is used to avoid the need of manually setting the weights associated to each descriptor.

1 Introduction

In the recent years, the size of digital image collections has increased rapidly. This information has to be organized so as to allow efficient browsing, searching and retrieval. Increasing interest is being paid to the study of image retrieval. For that purpose, several strategies have been used including *manual annotations* and *Content-Based Image Retrieval (CBIR)* [3]. In CBIR, low-level features such as color, shape, position, texture are extracted for the images. Examples of systems that are, at least partially, based on CBIR include QBIC [1], Virage, VisualSEEk [7] and MARS [4].

Most of the literature on CBIR focuses on the characterization and retrieval of individual images or regions. The goal of this paper is to propose a framework for the retrieval of multiple regions characterized by low-level features. The retrieval process combines the assessment of the similarity between regions (called region similarity) and of the similarity of the relationship between these regions (called structural similarity). In this context, one of the first issues to be faced is the selection of the scale at which the description has to be done. Since the queries are unknown and may deal with very different scales, it is not pertinent to fix a priori the description scale. As a result, a multiscale representation should be used. In this paper, we propose the Binary Partition Tree [6] as a basis for the image description. Finally, relevance feedback is used to avoid the need of manually setting the weights associated to each descriptor.

The organization of this paper is as follows: Section 2 gives an outline of the framework. The strategy used to search for multiple regions is described in sec. 3. The relevance feedback mechanism is discussed in sec. 4. Finally, results are reported in sec. 5.

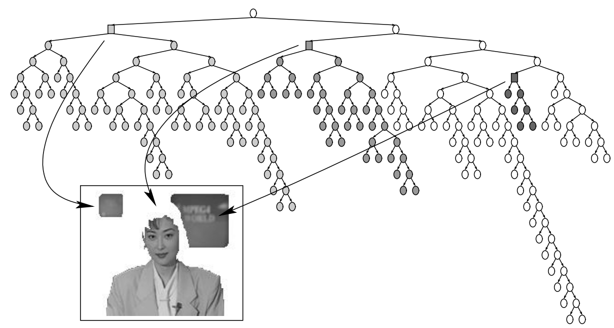

Figure 1: Example of BPT and of multiple regions retrieval.

2 Outline of the strategy

In our work, images in the database are represented by means of Binary Partition Trees (BPT) [6]. A BPT represents a hierarchical set of partitions of an image. The leaves of the tree represent small regions and the remaining nodes represent regions obtained by merging their children. The root node represents the entire image support. As can be seen in Fig. 1, the BPT is a hierarchical region-based representation of the image where hierarchy is created by the inclusion relationship between regions. There exist efficient algorithms to compute BPTs [6].

Once a BPT has been created, each of its node is described by a set of low-level descriptors. Any set of low-level descriptors can be used, but in our work we have used the following:

- Mean color in the YUV. Note that, because of the BPT structure, a region color can be described using only the descriptor of its node but a more precise description can also be reached when a region is described using the leaf nodes of the associated sub-tree. As a result, the color of each region is characterized by a histogram.
- Position, size and orientation extracted from a principal component analysis of the region support.
- Shape characterized by the Curvature Scale Space (CSS) descriptor [2]. This descriptor characterizes the points of high curvature along the region contour.

The query is made of an arbitrary number of regions. The user can select the various weights associated to the region descriptors (region similarity) as well as the weight associated to the similarity in terms of region relative position (structural similarity). As the user may have difficulties in appropriately defining the weights, a relevance feedback mechanism can be used as discussed in 4. Finally, the search engine interacts with the database in order to find visually similar objects with respect to the query. That is the search engine selects, among the set of nodes of the trees stored in the database, those whose contents are similar to those of the query. Fig. 1 shows an example of multiple region retrieval.

3 Multiple regions search

For notation purposes, let us denote by $Q = \bigcup_k Q_k$ the query which is made up of the regions Q_k. Let us also denote by $T = \bigcup_k T_k$ the targets in the database. A simple searching solution would consist in generating, for each BPT in the database, all possible target objects T by generating all possible combinations of its nodes that matches the number of query regions. The target objects could then be ordered according to their similarity with respect to the query. This brute force solution is computationally very expensive and thus a sub-optimal algorithm has been developed. The approach to search for Q in a BPT stored in the database can be divided in three stages.

1) Retrieval of visually similar regions

The first stage obtains a list, L_1, of visually similar regions with respect the regions the query is composed of. The region similarity is computed as follows:

$$S_{region}(Q_i, T_j) = \sum_k \omega_{i,k} S_k(Q_i, T_j) \qquad (1)$$

where Q_i and T_j represent the regions of the query and the target, respectively. $S_k(Q_i, T_j)$ is the similarity between query and target using descriptor k (color, position, size, orientation and shape), and ω_k is the weight associated to descriptor k for region i.

Fig. 2 shows an example that will be used throughout this section. The BPT is made of 15 regions and the query object involves three regions. The BPT nodes, $T_1 \dots T_{15}$, are compared against the regions of the query: $Q_1 \dots Q_3$. In this example, T_5 is the region that is visually most similar to Q_2 among all possible combinations.

2) Reference region selection and query normalization

The objective of the second stage is to select a reference region in order to normalize the query, and to create a list, L_2, of regions ordered according to their region and structural similarity. The algorithm first extracts from the list L_1 the pair (Q_i, T_j) of most similar regions. The target region T_j is used as a reference to obtain the normalized position and size of the query regions. This normalization of the query is necessary to be able to measure the structural similarity. The normalization approach is illustrated in Fig. 3. On the left, the query Q composed of

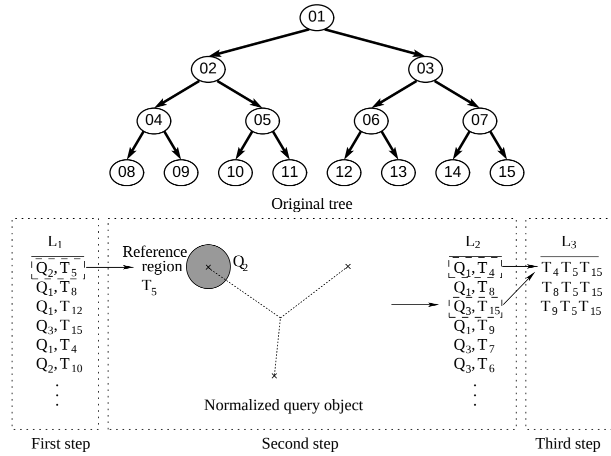

Figure 2: Example of multiple regions search in a BPT.

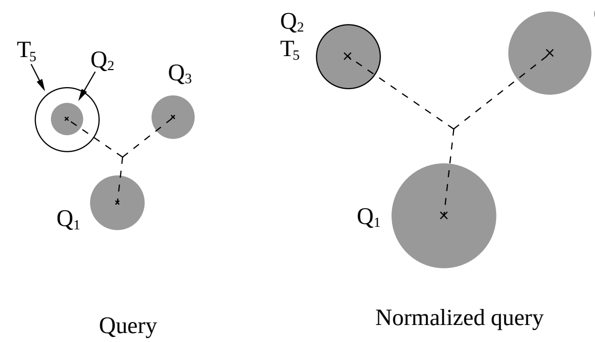

Figure 3: Example of query object normalization. Left: query object with three regions. The position and size of Q_1, Q_2 and Q_3 are normalized according to the ratio $Size(T_5)/Size(Q_2)$.

three regions is shown. Each region is represented with a node. Assuming, for instance, that the reference region is T_5 and that is corresponds to Q_2, the normalization step essentially scales the position and size of the query regions according to the difference in size between T_5 and Q_2. After normalization T_5 and Q_2 have the same size and the remaining query regions are scaled in size and position according to the ratio $Size(T_5)/Size(Q_2)$. The normalized position and size of the query regions, can interpreted as the "ideal" position and size of the target regions, if T_5 is used as reference.

Once the normalization has been performed, each pair $(Q_i, T_j)_{j \neq 5} \in L_1$, is analyzed to compute its structural similarity. The structural similarity $S_{struct}(Q_i, T_j)$ is composed of two terms: structural size and structural position similarity (see Fig. 4). The structural position similarity evaluates the difference between the "ideal" query region position (normalized Q_i) and the regions of the BPT (T_j). This difference $\varepsilon_{pos}(Q_i, T_j)$ is basically the Euclidean distance between the regions center of gravity.

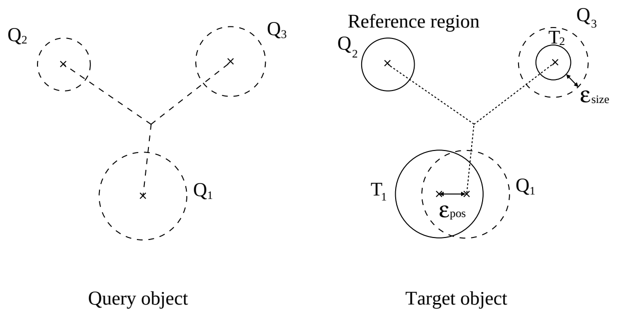

Figure 4: Assessment of structural similarity: The structural position error (ε_{pos}) and structural size (ε_{size}) errors are illustrated for target regions T_1 and T_2, respectively.

The structural size similarity assesses the difference in size between regions: $\varepsilon_{size}(Q_i, T_j) = \frac{\bigvee\{Size(Q_i), Size(T_j)\}}{\bigwedge\{Size(Q_i), Size(T_j)\}} - 1$.

Finally, the structural similarity between regions Q_i and T_j is given by: $S_{struct}(Q_i, T_j) = \omega_{stpos}\varepsilon_{pos}(Q_i, T_j) + (1 - \omega_{stpos})\varepsilon_{size}(Q_i, T_j)$, where ω_{stpos} is the weight given to the structural position error. Finally, the overall similarity between regions Q_i and T_j is obtained as a weighted sum of its region and structural similarity:

$$\begin{aligned} S_{overall}(Q_i, T_j) &= (1 - \omega_{struct})S_{region}(Q_i, T_j) \\ &+ \omega_{struct}S_{struct}(Q_i, T_j) \end{aligned} \quad (2)$$

where ω_{struct} ($0 \leq \omega_{struct} \leq 1$) is the weight associated to the structural error. The weight ω_{struct} allows the user to indicate the relative importance of the spatial relationship of the regions composing the target compared to their low-level visual characterization. A second ordered list, L_2, is used to order the pair (Q_i, T_j) following the overall distance $S_{overall}(Q_i, T_j)$.

In addition to the pairs (Q_i, T_j) stored in L_2, pairs representing missing regions are also inserted in L_2: for each target region T_j, an empty pair is constructed, $(Q_i, \emptyset)$, and inserted at the position given by ξ_h. ξ_h may be considered as the penalty corresponding to the absence of a target region with respect to the query. As a result the retrieved targets do not have necessarily the same number of regions as the query.

3) Retrieval of multiple regions

In the third and last stage, the list L_2 is analyzed in order to obtain a set of meaningful target object T. At this point only one region of T is known, namely the reference region T_j found in second stage. The algorithm described next is an iterative algorithm that finds the remaining target regions. In the example of Fig. 2, we know that T_5 is the reference region (corresponding to Q_2) and we are looking for two remaining regions that are similar respectively to Q_1 and Q_3. For that purpose, the list L_2 is analyzed by extracting first the most similar pairs and we construct the triplet of target regions by selecting the best regions corresponding to Q_1 and Q_3. In our example, the pairs (Q_1, T_4) and (Q_3, T_{15}) are selected to create the target $\{T_4, T_5, T_{15}\}$. The final similarity between this target and the query is given by sum of the overall similarity (Eq. 2) for the corresponding pairs of regions. This similarity value is used to insert the proposed target object, T, into a list L_3 which orders the target objects found for all the trees in the database.

The present algorithm only extracts one target T per BPT. Several other targets are extracted by introducing the following modifications: 1) in the second stage, reference region not only the first T_j of L_1 is considered but also the second or the third one; 2) in the third stage, while looking for target regions T_j corresponding to the query region Q_i not only the very best candidates are selected but also the second or the third candidate. As can be seen in Fig. 2, the proposed search algorithm is able to construct several meaningful target objects for a fixed tree. The whole process is repeated for each tree included in the database. At the end of the search, the list L_3 stores, with increasing order of distance, a set of target objects visually and structurally similar to the query.

4 Relevance feedback

The retrieval algorithm discussed in the previous section involves a set of weights to control the relative importance of the various features. The usefulness of this approach is limited if the end user is actually asked to manually tune the weights to optimize the retrieval process. One possible solution to this problem is the so-called relevance feedback [5]. The user only needs to mark the set of images he or she thinks are relevant to the query. The weights embedded in the query object are then automatically updated to model the high level concepts and perception subjectivity.

Let us consider Eq. 1. The weights ω_k are associated to the similarity value $S_k(Q_i, T_j)$ of the feature k. An initial search is performed with an arbitrary value of the weights (for example, all weights can have the same initial value if no feature is clearly more relevant for a given retrieval task). The retrieved results are shown to the user who select those objects T_n he agrees with according to his information need and perception subjectivity. We assume that the selection is binary: either select or do not select the result as relevant. The weights can then be computed by the following rule:

$$\omega_{i,k} = 1/\sum_{\{T_n\}} S_k^2(Q_i, T_n) \quad (3)$$

where $\{T_n\}$ is the set of target marked as relevant by the user. Intuitively, if all the relevant objects have low distances for the feature k, it means that feature k is a good indicator of the user information need. On the other hand, if the distance values for feature k are very different or have high values among the relevant objects, then feature k is not a good indicator. The values ω_k are then normalized so that $\sum_k \omega_k = 1$. The same approach is used for the weights appearing in Eq. 2.

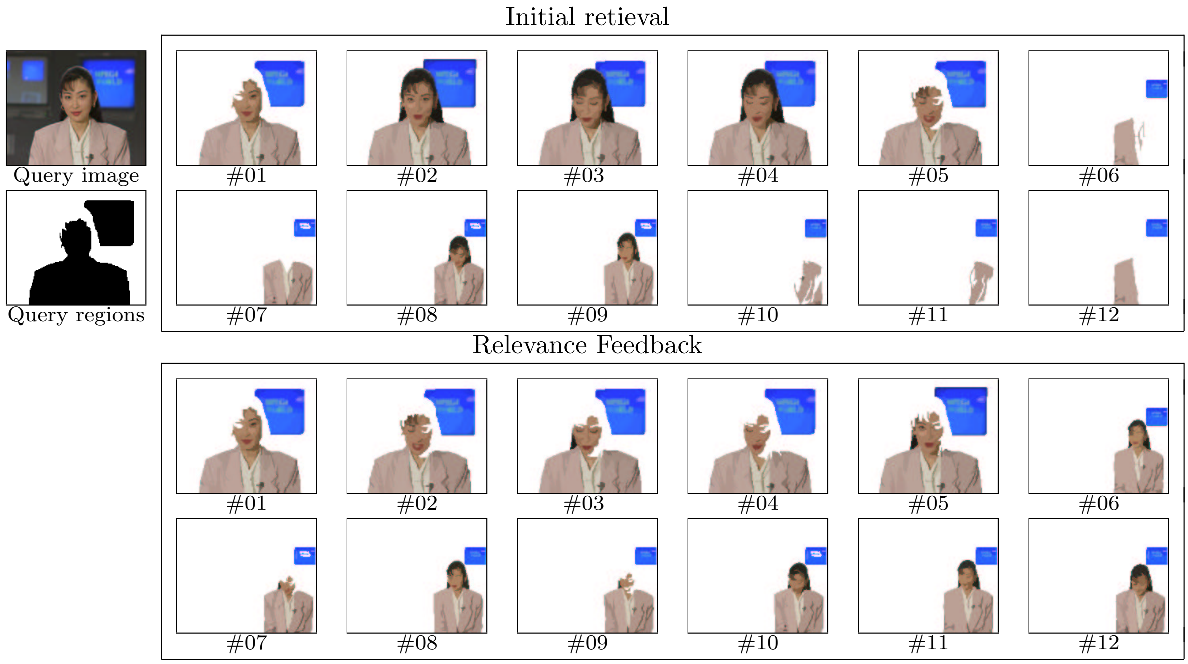

Figure 5: Retrieval results for the query on the left. Top: retrieved regions with the following weights: $\omega_{color} = \omega_{shape} = 0.5$, $\omega_{size} = \omega_{pos} = 0$ and $\omega_{struct} = \omega_{struct} = 0.5$; Bottom: retrieved regions after relevance feedback.

5 Discussion

An illustration of this framework for the retrieval of multiple regions is given in Fig. 5. The image and regions of the query are shown on the left side of the Figure. The query is made up of two regions: the newscaster and a blue screen. The database has been constructed by computing the BPT and the corresponding descriptors for a set of images extracted from various video sequences. Weights have been set as follows: $\omega_{color} = \omega_{shape} = 0.5$, $\omega_{size} = \omega_{pos} = 0$ and $\omega_{struct} = \omega_{stpos} = 0.5$. Note that in this case, the structural relationship, color and shape are considered as important.

The retrieved items are shown on the top of Fig. 5. As can be seen, items #1 to #12 correspond to targets that are visually and structurally similar to the query. Note that the search engine is able to retrieve objects independently of their size with respect to the query region ($\omega_{size} = 0$), see for instance items #6 to #12.

An example of relevance feedback is shown on the bottom of Fig. 5. From the retrieved items presented on the top of the figure, items #1 to #4, #8 and #9 are selected as relevant by the user. The search engine then sets automatically the weights as follows: for both regions, $\omega_{color} \approx 1$, $\omega_{shape} \approx \omega_{size} \approx \omega_{pos} \approx 0$, $\omega_{struct} \approx 0.2$ and $\omega_{stpos} = 0.92$. Observe that the search engine is able to retrieve a higher number of relevant results.

In this framework for multiple region retrieval, the BPT has been used since the tree representation offers a way of representing image regions at different scales of resolution. To use such representations for region-based retrieval, region descriptors have to be attached to the BPT nodes. Our work has focused only on low level descriptors. The retrieval is based on the visual similarity between regions but also uses the spatial organization of the regions composing the searched object.

References

[1] M. Flickner, et al. Query by image and video content: the QBIC system. *IEEE Computer*, 28(9):23–32, Sept. 1995.

[2] F. Mokhatarian, A.K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. on PAMI*, 14(8):789–805, Aug. 1992.

[3] Y. Rui, T.S. Huang, S.-F. Chang. Image retrieval: current techniques, promising directions and open issues. *Jour. of Visual Com. and Image Rep.*, 10:39–62, March 1999.

[4] Y. Rui, T.S. Huang, S. Mehrotra. Content-based image retrieval with relevance feedback in MARS. In *IEEE Int. Conf. on Image Proc.*, Santa Barbara, USA, Oct. 1997.

[5] Y. Rui, T.S. Huang, M. Ortega, S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Trans. on Circuits and Syst. for Video Tech.*, 8(5):644–655, Sept. 1998.

[6] P. Salembier, L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. *IEEE Trans. on Image Processing*, 9(4):561–576, Apr. 2000.

[7] J.R Smith, S.-F. Chang. Visualseek: a fully automated content-based image query system. In *ACM Multimedia*, Nov. 1996.