

NEW IMAGE RETRIEVAL PARADIGM : LOGICAL COMPOSITION OF REGION CATEGORIES

Julien Fauqueur and Nozha Boujemaa

INRIA, Imedia Research Group, BP 105, F-78153 Le Chesnay, France

ABSTRACT

We present a novel framework for intelligent search and retrieval by image content composition. Very different from the existing Query-by-Example paradigm, logical queries are expressed using categories of similar regions **without any starting example region**. The set of region category representatives constitutes the “photometric region thesaurus” of the image database.

Logical composition of region categories expresses the presence and absence of certain types of regions in images to retrieve and allows to integrate visual semantics in the search.

Resulting indexing and retrieval implementation turns out to be simple and very fast even on very complex query compositions and large image database. It was tested on a database of 9,995 images from the Corel Photostock.

1. INTRODUCTION

The earliest Content-Based Image Retrieval approach is the global query-by-example approach, such as QBIC [1] system. This approach provides approximate results when the focus of the search is a specific object or part in an image. Partial query formulation allows the user to specify which part of the image is the focus of his interest and leads to a higher user satisfaction. Partial query systems based on image regions have been proposed to allow more specific queries. Among the few existing region-based query systems we can cite BLOBWORLD[2], NETRA[3] and more recently IKONA[4]. These systems simply perform an *exhaustive* search among regions in database from a single example region. In VISUALSEEK[5], a multiple region query was proposed by sketching rectangles of synthetic colors and synthetic textures.

Be they global, single or multiple region based, existing CBIR systems all rely on the same paradigm : query-by-example (example being an image or one or more regions) and retrieval by exhaustive search. This approach is well suited to perform visual comparison between a given example and entries in the database, i.e. to answer to such a query : “find images/regions in the database similar to this

image/regions”. But very often, the user doesn’t have an example image to start the search. **The target image is only in his mind**. In this case, the prior search of an example to perform the actual query by example is tedious, especially in the case of a multiple region query.

The new framework presented in this paper differs completely from this paradigm on both query and retrieval processes. After segmenting images in the database, all extracted regions are grouped into categories of visually similar regions. In the query interface, the set of category representatives provides an overview of the types of regions which constitute the images in the database. They can be viewed as a “photometric region thesaurus”. Images are simply indexed by the list of category labels and, as a consequence, the user can very quickly retrieve images from logical queries as complex as : “find images composed of regions of these types and no regions of those types”.

In section 2, we will briefly present the algorithm to group visual features. In section 3, the generation of region categories and their neighbor categories to allow range queries will be explained. Then, in section 4, we will detail the approach to achieve image retrieval by composition of region categories. We will present an adapted retrieval strategy to handle queries from complex compositions. In section 5, we will present an original user interface along with the results. In section 6 concluding remarks on the specificity of this new query by composition framework and future work will be addressed.

2. VISUAL FEATURE GROUPING METHOD

To generate region categories, an efficient clustering scheme is required. The CA (*Competitive Agglomeration*) clustering [6] was chosen because of its major advantage to determine automatically the number of clusters. Using notations from [6], we call $\{x_j, \forall j = 1, \dots, N\}$ the set of data to cluster and C the number of clusters. $\{\beta_i, \forall i = 1, \dots, C\}$ denote the prototypes to be determined. The distance between data x_j and prototype β_i is $d(x_j, \beta_i)$. The CA-clustering is performed by minimizing the following quantity J :

$$J = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^2 d^2(x_j, \beta_i) - \alpha \sum_{i=1}^C [\sum_{j=1}^N u_{ij}]^2 \quad (1)$$

Subject to membership constraint: $\sum_{i=1}^C u_{ij} = 1, \forall j = 1, \dots, N$, where u_{ij} represents the membership degree of feature x_j to cluster i . Minimizing the first term provides the C prototypes and the fuzzy partition U which minimize distances between data and prototypes. Minimizing the second term agglomerates clusters. So minimizing J with an over-specified number of initial clusters classifies data and simultaneously optimizes the number of classes. The clustering granularity is controlled by factor α . CA is used in our framework to group similar regions.

3. CATEGORISATION AND RANGE QUERY IN THE REGIONS FEATURE SPACE

To extract regions, we adopt the image segmentation technique presented in [4]. It was developed specifically for image retrieval by regions. It is based on the CA-clustering of Local Distributions of Quantized Colors (*LDQC's*).

We define the **region categories** (denoted C_1, \dots, C_P) as the clusters of regions which have similar visual features. They are the basis of the definition of similar regions in the retrieval phase. Here we choose to characterize regions with their mean color such that regions from the same category have similar mean color. It is important to note that other visual cues could be used such as color distribution, position, surface. Despite the straightforwardness of mean color description, we will see it is sufficient to form generic categories. Regions mean colors are determined in the LUV space, which is chosen for its perceptual uniformity. In the color space, all the regions mean colors form a compact and relatively dense data set and no natural data grouping can be expected. We can not make a priori assumption concerning the well-definition of clusters of regions for any database. But what can always be guaranteed is an intra-category homogeneity by setting a fine clustering granularity.

Region categories are formed by grouping the regions mean color features with CA and a fine granularity. For each region category, its **representative region** is defined as the closest region to its prototype. Representative regions are used to identify each category in the query interface.

Since similarity between regions will be defined, at a first level, as members of the same category, a fine clustering granularity will ensure the retrieval of very similar regions (hence high retrieval precision). At a second level, we also consider as similar regions which are in close categories (called “neighbor categories”) to also allow high recall. This key idea allows to achieve **range queries** in the regions feature space. The **neighbor category** of a category C_q of prototype p_q is defined as a category C_j whose prototype p_j satisfies $\|p_q - p_j\|_{L^2} \leq \gamma$, for a given range radius threshold γ . Range radius γ is adjusted at the query phase. We call $N^\gamma(C_q)$ the set of neighbor categories of a category C_q . See figure 1 for an illustration of the definition of neigh-

bors using the radius. Note that thanks to this range query scheme, the search is less dependent on the database partition into categories since all close categories are considered together.

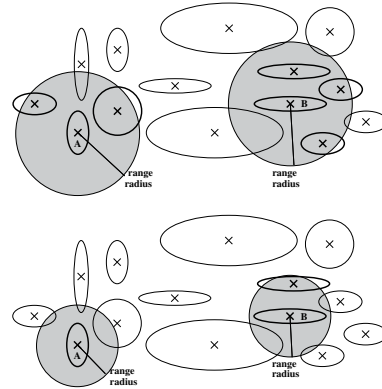


Fig. 1. Range radius and neighbor categories. A and B are two categories. Neighbor categories are drawn with thick contours. Prototypes are identified by crosses. The grey disks of radius γ cover the neighbor categories $N^\gamma(A)$ and $N^\gamma(B)$. A high radius (top) or a lower radius (bottom) integrates more or less neighbor categories to define the type of searched regions.

The combination of homogeneous region categories with the integration of neighbors categories is the key choice in the definition of the range query scheme.

4. IMAGE RETRIEVAL BY COMPOSITION

From this point on, regions aren't considered individually anymore but are totally identified to the category they belong to. With the help of all categories representative regions, the user will select Positive Query Categories (referred to as **PQCs**) and Negative Query Categories (**NQCs**). The PQCs correspond to the user-selected categories of regions which should appear in retrieved images. They are denoted as $\{C_{pq_1}, \dots, C_{pq_M}\}$. The NQCs correspond to the user-selected categories of regions which should *not* appear in retrieved images and are denoted as $\{C_{nq_1}, \dots, C_{nq_R}\}$.

In its most complex form, a **query composition** is defined as the formulation: “find images composed of regions in these PQCs and no region from those NQCs”. It is expressed as the list of PQC labels $\{pq_1, \dots, pq_M\}$ and NQC labels $\{nq_1, \dots, nq_R\}$.

Performing a query composition first requires to retrieve images which contain a region from a single PQC category denoted C_{pq} say. For a given category C_{pq} , we define $IC(C_{pq})$ to be the set of images containing at least one region belonging to category C_{pq} . To expand this search to a range query, we now take into account neighbor categories of C_{pq} by defining relevant images as those which have a

region from category C_{pq} or from any of its neighbors:

$$\bigcup_{C \in N^\gamma(C_{pq})} IC(C) \quad (2)$$

Range radius threshold γ is set in the user interface.

To extend the query to all M PQC: $C_{pq_1}, \dots, C_{pq_M}$, we search images which have a region in C_{pq_1} or its neighbors and ... and a region in C_{pq_M} or its neighbors. The set S_Q of images satisfying this multiple query is now written as:

$$S_Q = \bigcap_{i=1}^M \left[\bigcup_{C \in N^\gamma(C_{pq_i})} IC(C) \right] \quad (3)$$

Then, to also satisfy the negative query we must determine images which contain a region from any of the R NQCs: $C_{nq_1}, \dots, C_{nq_R}$. As before, neighbor categories should also be taken into account. So the set S_{NQ} of images containing the NQCs is written as:

$$S_{NQ} = \bigcap_{i=1}^R \left[\bigcup_{C \in N^\gamma(C_{nq_i})} IC(C) \right] \quad (4)$$

The set S_{result} of retrieved images which have regions in the different PQC and which don't have regions in the NQC is expressed as the set subtraction of S_Q and S_{NQ} :

$$S_{result} = S_Q \setminus S_{NQ} \quad (5)$$

This set S_{result} constitutes the final set of relevant images.

Unions, intersections and subtractions in the expression of S_{result} are directly equivalent to formulate the query with logical operators as illustrated in figure 3: OR between the neighbors (expression 2), AND between query categories (exp. 3), ANDNOT for negative query categories (exp. 5).

To evaluate the expression of S_{result} (5), the brute force approach would consist in testing, for each image in the database, if it contains regions belonging to the PQC (and their neighbors) but contains no region in any of the NQC (and their neighbors).

Instead, to reduce dramatically this number of tests in a simple way, we use the fact that S_{result} is expressed as intersections and subtractions of image sets. The idea is to initialise S_{result} with one of the image sets and then discard images which don't belong to the other image sets. This initialisation avoids testing individually each image of the database. We directly start off with a set of potentially relevant images. S_{result} will be gradually reduced as follows :

1. initialise S_{result} as the set $\bigcup_{N^\gamma(C_{pq_1})} IC(C)$.
2. discard images in S_{result} which do not belong to any of the other union categories ($i = 2, \dots, M$) to obtain the intersections of S_Q (expression 3). At this point, we have $S_{result} = S_Q$.
3. to perform the subtraction of S_{NQ} from S_{result} , discard in S_{result} images which belong to any of the negative-query union categories ($i = 1, \dots, R$). We get $S_{result} = S_Q \setminus S_{NQ}$ (exp. 5).

Gradually, S_{result} is reduced from $\bigcup_{N^\gamma(C_{pq_1})} IC(C)$ to $S_Q \setminus S_{NQ}$. By this approach, we'll see in next section that a significant fraction of the database is not accessed at all.

This retrieval scheme is easily implemented using three hash tables which provide associations between categories, neighbor categories and images. It is important to note that at retrieval time we don't deal with regions themselves but only with images and labels of region categories, so that we don't have to individually access the large number of regions in the database. Search process is very fast since it only involves elementary operations on integers, unlike classic search approaches which require distance computations between multidimensional feature vectors.

5. RESULTS AND USER INTERFACE

Our system was tested with a 498 MHz Pentium PC. The test database consists of 9,995 images from Corel Photostock. 50,220 regions were automatically extracted from the 9,995 images. Clustering the 50,220 regions mean colors takes 150 seconds with CA. 91 categories are automatically generated and their populations range from 112 regions to 2048 regions. Since data are dense in the color space and the granularity is fine, CA algorithm provides categories homogeneous in mean color as expected. Intra-category variability is mainly due to different textures which have the same mean color.

Mean color is sufficient to define coarse types of regions in our approach. For specific applications or to discriminate regions more precisely, other regions descriptors can be used. The evaluation does not aim here at investigating the retrieval precision but at showing the viability of this new approach. Note that if other descriptors are used, categories are expected to group regions of similar textures, similar geometrical features, ...

As illustrated in figure 2, the query interface presents to the user the 91 category representatives of the categories, which provide an overview of the available types of regions in the database. Each representative is used to select corresponding category as a PQC or NQC. No browsing is required to pick any example image. Any combination of query composition can be expressed from this interface. In the query window, the range box allows to adjust the range radius γ which defines interactively the neighbor categories.

Query interface works like the "photometric region thesaurus" of the database. Its combination with the query by logical composition makes this system similar to a text retrieval one, with images as text, regions as words, categories as concepts and neighbor categories as similar concepts.

The relevance of matched regions relies on the region extraction and grouping schemes. In retrieved images, regions from the positive query categories are salient. From the user point of view, false positives among matched re-



Fig. 2. In the query interface, the user can select each of the 91 category representatives as a PQC or a NQC. This interface constitutes the “photometric region thesaurus”. (full image at [7]).



Fig. 3. After the user selection of 2 PQCs and 1 NQC and the range value, the system translates the query into corresponding logical composition of region categories in which neighbor categories are separated by disjunctions.

regions are few and correspond to hard segmentation cases in complex composite natural images. Concerning the precision of composition matching in retrieved images, the simplicity of the indexing and retrieval scheme (comparison of category labels in image indexes) ensures a high user satisfaction. Indeed, in retrieved images, salient regions do satisfy the constraints of presence of regions from PQCs and the absence of regions of NQCs. Query can be refined by adjusting the range radius to widen or narrow the types of searched regions by considering more or less neighbors.

Consider an example of query composition. In the Corel database, to search cityscapes, the user will search images with a building, some sky and no vegetation. This can be translated into the following query composition : “images composed of a grey region and a blue region but no green region”. Figure 3 illustrates this query. Figure 4 (cropped screenshot) shows the relevant set of images retrieved for this query among the 9,995. In these images, grey regions match buildings, monuments or rocks and blue regions essentially match sky. Many nature landscapes are rejected by the constraint of absence of green region (figure 5).



Fig. 4. Retrieved images from query composition “grey region and blue region and not green regions”. Images do contain a blue and grey region but no green region. (full image at [7]).

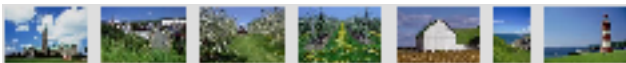


Fig. 5. Images rejected from the “cityscape” query due to the presence of a green region.

The image retrieval scheme is very fast first because it

only involves accesses to hash tables (no feature vectors nor distance are involved) and second because it is *not region exhaustive* (we only deal with the 91 regions categories instead of the 50,220 regions) and third because it is *not image exhaustive* (only a fraction of the image database is accessed). On average on various query compositions, the fraction of accessed image entries is around 12%. Retrieval process takes up to 0.03 second for the most complex queries.

6. CONCLUDING REMARKS

We have presented a framework to retrieve images based on logical composition of region categories. The system allows retrieving images by query composition like : “find images with regions of these types and not like those types”. The originality of this approach relies on the grouping of similar regions into categories and has the following advantages:

- no required starting example region
- query by image composition using regions categories
- natural region range query by interactive definition of neighbors categories
- efficient indexing and very fast image retrieval

Although a very simple color region feature is used, the constraint of composition in retrieved images seems to express some underlying “visual semantics” in images.

This framework can lead to further developments such as proposing a more perceptual arrangement of categories in query interface, integrating other region descriptors, developing a hierarchical region categorization to handle very large databases.

7. REFERENCES

- [1] M. Flickner and al., “Query by image and video content: the qbic system,” *IEEE Computer*, 1995.
- [2] C. Carson and al., “Blobworld: A system for region-based image indexing and retrieval,” *Proc. of Intl. Conference on Visual Information System*, 1999.
- [3] W. Y. Ma and B. S. Manjunath, “Netra: A toolbox for navigating large image databases,” *Multimedia Systems*, 1999.
- [4] J. Fauqueur and N. Boujemaa, “Region-based retrieval: Coarse segmentation with fine signature,” *Intl. Conference on Image Processing (ICIP)*, 2002.
- [5] S.F. Chang J.R. Smith, “Visualseek: A fully automated content-based image query system,” in *ACM Multimedia Conference, Boston, MA, USA*, 1996.
- [6] H. Frigui and R. Krishnapuram, “Clustering by competitive agglomeration,” *Pattern Recognition*, 1997.
- [7] www-rocq.inria.fr/fauqueur/COMPOCAT/