STRICT: An Image Retrieval Platform for Queries Based on Regional Content

Jean-Francois Omhover and Marcin Detyniecki

LIP6 – Pole IA
Universite P. et M. Curie - CNRS
8 rue du Capitaine Scott
75015 Paris

Abstract. This paper presents a CBIR system that is based on a segmented representation of image content. It compares regional features using fuzzy similarity, which have been shown to be psychologically intuitive. We show that they can be aggregated to support four different type of original queries. The system also supports competitive queries to test different visual comparison measures, and lets the expert user manipulate the parameters of the functions involved.

Keywords: Image Retrieval, Segmentation, Fuzzy Similarity Measures, Aggregation.

1 Introduction

In the last decade, there has been an increasing interest in image retrieval systems. The idea is to let some user query an image database for a specific image content. Unfortunately, this common goal has not yet been achieved [6].

The community has focused on the automatic extraction and comparison of image features. The query is then expressed not as some textual description, but as an image, chosen by the user. As an answer, the user obtains the list of images in the database that are found most similar to his request. Processing such a comparison is based on a set of signatures extracted from the images and is associated to a similarity measure.

In the last few years, segmentation has been used as a tool for image processing. Images can be segmented in regions, roughly corresponding to objets in the image, thus reflecting the objective content of an image. Measures have been developped to compare regions one to the other.
For the moment, only a few image retrieval systems follow this approach. For instance, Blobworld [1] proposes its segmentation tool to the user, who specifies the request by selecting a set of regions of interest. The retrieval then consists in comparing pairs of regions. The returned images are those which obtained the best scores in the one-to-one region similarity computation. Simplicity [7,2], another system, extends this to image-to-image comparison. It compares images globally by aggregating region-to-region similarities. Unfortunately, its scheme

P. Enser et al. (Eds.): CIVR 2004, LNCS 3115, pp. 473–482, 2004.

does not include the spatial configuration of the regions, though it tries to match regions of the request to regions of each image.

In this paper, we present a system called "STRICT". It has been designed to be used as a platform for testing visual similarity measures. It is open to the implementation of new similarity measures. It lets the user modify its parameters online. It is also capable of running several requests in parallel, to compare their effectiveness. It proposes a score-post-processing tool to aggregate the result of different comparison measures. Based on these tools, we propose four original visual requests using the aggregation of global or regional similarity measures.

In the following, we present the architecture of STRICT. In section 3, we introduce the regional features that were implemented. In section 4, we propose four original type of visual requests our system supports. Finally, in section 5, we illustrate our approach with some experimental requests.

2 STRICT Image Retrieval Platform

We developped a complete image retrieval system called STRICT [8]. This system has been designed so that it fulfills the five following constraints. First, it proposes image retrieval capabilities based on regions. Secondly, the features of these regions are extracted automatically, in an offline process. Third, as there are many parameters which can be adapted to improve the effectiveness of a visual comparison measure, the system lets the user modify these parameters and instantly run the request using these new values. Four, it also proposes a score post-processing tool to aggregate different measures of similarity. Finally, it runs parallel requests, enabling the user to get the result using different similarity functions, and compare them. In this section, we expose the architecture of STRICT. We also detail the request protocol.

2.1 Indexation Server

Our architecture is composed of two distinct systems. The first is our own indexation server that keeps the signature database and computes similarities on demand. The second is an HTTP server that runs the interface, serves the images to the user and presents the results of his requests.

The indexation server remains invisible to the user. It maintains a database of vectors extracted from an images set. These vectors are extracted offline before launching the server. They can be of any type: either global (like an histogram or color moments), regional (the image is first automatically segmented then features are extracted from each region), or even textual (if annotations are given with the image set).

For each indexation process (global, regional, textual), the server maintains the set of features, extracted from each image. Associated to each index, the system keeps a register of similarity measures. Each of these measure is adapted to the index it concerns.

These measures may have parameters. In the register, the server also keeps a list

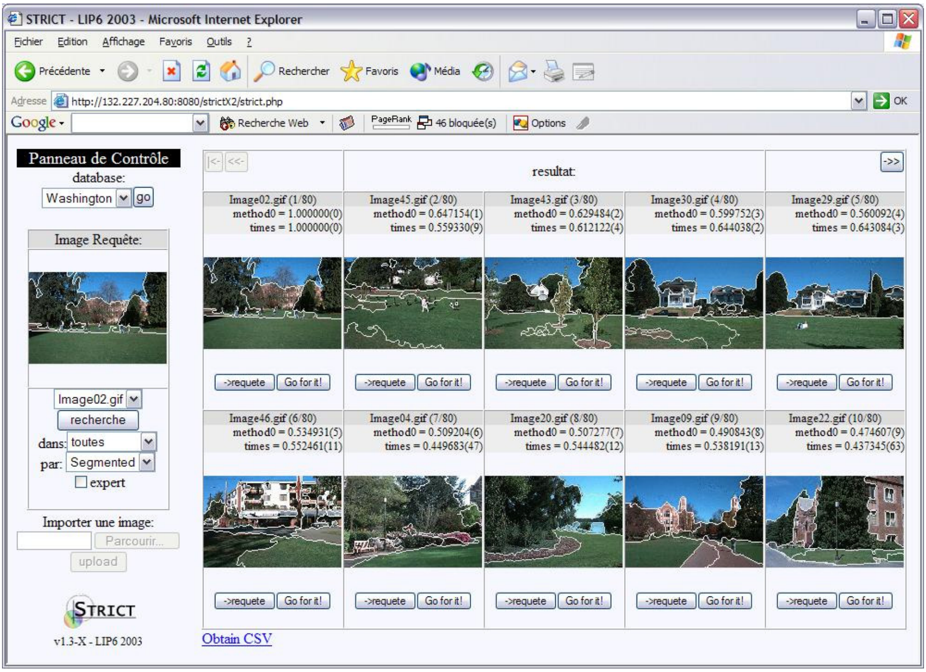

Fig. 1. STRICT Interface, browsing a request's result

of the type, the possible values, and the description of each these paramaters. It also contains a list of mathematical operations that can be used to build a post-processing function.

2.2 Web Interface

The indexation server receives the queries from the separate interface. It is a web interface which is the visible part of the system. Its dynamic format enables the expert-user to browse the image database (see figure 1), formulate his queries and see the results using the different similarity measures maintained by the indexation server. When an user opens a new connection to the interface, the description of each measure is sent from the indexation server to the web server. From this list, the web server generates a form for the user that allows to modify the parameters. Thus the interface has not to be modified to cope with modifications or additions of comparison measures or of new parameters.
Using the interface, the user can then compose his request with several measures with different parameters. He can choose to run multiple measures. Each measure of the request, when received by the indexation server, is then run against every image of the database. Each score, resulting from each measure specified by the user, is sent back to the interface. When the resulting list of images is displayed, each image is marked by the intensity of each measure specified. This

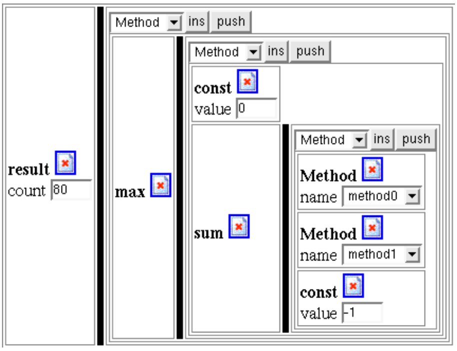

Fig. 2. Lukasiewicz t-norm $[(x, y) \mapsto max(x + y - 1, 0)]$ built by simple clicks on STRICT web interface

feature guides the user in the comparison of the effects of its parameters on the retrieval.

The list of the measures involved are not the only information sent in the query. The system can also aggregate these results on-the-fly. The user can formulate an aggretation operator using basic mathematical operations programmed in the indexation server. Using the minimum, the maximum, the addition and the product, he can define his own post-processing score operator (see figure 2). By coupling a list of similarity measures with a post-processing operator, he creates a new measure that he can test against the images. The basic operations programmed in this application let us build most of the known aggregation operators [4]. And it is this feature that let us build interesting visual queries, as shown in section 4.

3 Segmented Image Vectors

From an image, a segmentation algorithm extracts regions complying to a given homogeneity criterium. This criterium is usually based on colors: the algorithm tends to isolate regions of connected pixels, which present similar colors. Those regions roughly correspond to the objects present in each image. A similarity measure based on those regions should reflect an objective similarity of the content.

In order to build a semantic similarity between images, we developed a similarity measure for regions based on the similarities of color, shape and position [8]. Here, we briefly present the segmentation algorithm. Then we introduce the definition of fuzzy similarity measures. And finally, we explain which region features we chose to extract and how these features are compared using fuzzy similarities.

3.1 Segmentation

For the segmentation of an image in regions, many different approaches exist [3]. They have been developped for thirty years and applied to various application fields. They all aim at building a crisp partition of the image, based on vectors computed from the pixels: color, texture coefficients, edge orientation.
Our algorithm [5] provides good results in a very short computation time. It follows the merge approach: each pixel is first considered as an isolated region, then fusions are operated to merge connected pixels of similar colors, until there is no more possible fusion. This arrives when all the connected regions are dissimilar enough.

3.2 Fuzzy Similarity Measures

In [13], Tversky introduces a new definition of similarity measure that breaks with the classical notion of distance. As shown in [11], these measures provide an intuitive measurement of the similarity. They are also independant on the scale of the feature sets. Furthermore, they provide a comparison score that is normalized to [0,1], so that it is easy to aggregate. For these three reasons, we implemented them in STRICT to compare the regional features.
Based on a psychological approach, Tversky's work proposes to evaluate the similarity between two sets of binary features by measuring their common and distinctive features. An extension of this definition to fuzzy sets can be used to compare sets of gradual features. Applied to the histograms H_A,H_B of two images A,B, this leads to the following computations:

- $M(A \cap B) = \sum_{C_i} min(H_A(C_i), H_B(C_i))$, the area of the features that are common to A and B.
- $M(A - B) = \sum_{C_i} max(H_A(C_i) - H_B(C_i), 0)$, the area of the features that dinstinguish A from B.
- $M(B - A) = \sum_{C_i} max(H_B(C_i) - H_A(C_i), 0)$, the area of the features that dinstinguish B from A.

According to Tversky, the similarity between A and B is a functions as these three areas. It is to notice that the classical histogram intersection proposed by Swain and Ballard [12] and all the successors are particular cases of this framework. Here, we chose to implement three other: Jaccard, Dice and Ochiai [10]. With $X = M(A \cap B)$, $Y = M(A - B)$, $Z = M(B - A)$, we have:

$$\begin{aligned} S_{jaccard}(X, Y, Z) &= \frac{X}{X+Y+Z} \\ S_{dice}(X, Y, Z) &= \frac{2X}{2X+Y+Z} \\ S_{ochiai}(X, Y, Z) &= \frac{X}{\sqrt{(X+Y)}\sqrt{(X+Z)}} \end{aligned}$$

The fact of considering different measures enable us to discover interesting properties. In particular, we show in [9] that the result ranking list provided by an information retrieval system is conserved in several cases.

3.3 Region Color, Shape, and Position

The ressemblance between regions is more elaborated than the global one. We have to consider several aspects of comparison, which are color, shape and position. For each of these comparisons, a region is represented as a vector. Because these three vectors belong to different spaces, and cannot be related one to the other, the similarity between pairs of region relies on three different measures: one measure for each of the vector pairs. For two regions $R(i)$ and $I(j)$, extracted respectively from the image request R and from an image database entry I, the region similarity S_{reg} can be written as a weighted mean of these three "sub-measures". With $\lambda_c + \lambda_s + \lambda_p = 1$:

$$\begin{aligned} S_{reg}(R(i), I(j)) = \lambda_c . S_{reg|color}(R(i), I(j)) \\ + \lambda_p . D_{reg|position}(R(i), I(j)) \\ + \lambda_s . S_{reg|shape}(R(i), I(j)) \end{aligned}$$

In our experiments, we have set the respective weights of these three measures to $\lambda_c = 0.6$, $\lambda_s = 0.2$, $\lambda_p = 0.2$.

Fuzzy Region Color Similarity The region color similarity $S_{reg|color}$ is based on the color histograms of the regions. To compare the regional color histograms, we use measures presented in section 3.2.

Proximity Measure The proximity measure $D_{reg|position}$ is the fuzzy inverse $(x \rightarrow 1 - x)$ of the normalized distance between the geometric centers of two regions. This distance is normalized so that the distance between the opposite corners of the images equals 1.

Fuzzy Shape Similarity The shape similarity measure $S_{reg|shape}$ is based on the centered binary masks of the regions. A mask is the matrix $K_{R(i)}$ of $\{0, 1\}$ that represents the crisp membership of each pixel of the image R to the region $R(i)$. $K_{R(i)}$ is centered so that its central point gives the membership of the geometric center of $R(i)$.
To compare the two shapes of regions $R(i), I(j)$, we compute their common and distinctive pixels, and apply the similarity measures of section 3.2.

4 Interesting Query Examples

STRICT can compute global and regional similarities. For the moment, the system is based on one-to-one comparison functions. For a single request image R, a similarity measure S is run against every entry I of our image database. The value $S(R, I)$ of the comparison is computed and returned for post-processing.
We adapt the global image approach to regions obtained using an automatic segmentation of the image. A regional similarity measure S_{reg} is then used to find in every image I a region that is similar to a single region $R(i)$ of image R. The returned score is noted $S_{reg}(R(i), I)$ and it corresponds to the truth value of the proposition *"there exists a region like $R(i)$ in I"*.
Those scores, corresponding to the similarity, can be aggregated on-the-fly under users' command. An operator, noted Agg, is used to procude a single value $f(I)$

from the n different similarities involved in the query. This feature enables the user to build different kinds of requests. This section points out four interesting schemes, where aggregation is the key element.

Multiple features: Using a single image request R, the user can compare the images on the basis of multiple features (color, texture...). STRICT then agregates the scores resulting from the different similarity measures. The final score $f(I)$ of the image I is then expressed as:

$$f(I) = Agg(S_1(R, I), ..., S_n(R, I))$$

where Agg can be the averaging operator, the minimum, the maximum, or any buildable operator (see section 2). For instance, the user can use a weighted mean to give more importance to the color against the texture.

Multiple image requests: The user can also specify multiple image requests. The idea is to retrieve images, which are similar to a set of examples (some of the examples might be more important than others). Based on $R_1, ..., R_n$, a set of request images, STRICT will compute the global similarity between each image I in the database and each R_i. It will then agregate these similarity to render a final score $f(I)$ for each I.

$$f(I) = Agg(S(R_1, I), ..., S(R_n, I))$$

The aggregation used must correspond to the goal of the request. For instance, setting Agg as the maximum will retrieve the images that are similar to *at least one* of the images $R_1, ..., R_n$. Instead, setting Agg as the minimum will retrieve images that are similar to *all* of the request images. In other words, the retrieved image has to be similar to each of the images of the request. Using the average will let the user weight some of the R_i, and therefore retrieve images that look like the strongly weighted images. In this last case, the user can also use a fuzzy negation applied to a similarity measure $S(R_i, I)$ to express that he wants images that does not look like R_i, and that are still similar to the other inputs.

Multiple regions: On a single image request R, the user can select different regions $R(i), ..., R(j)$ that he wants to look for in one image of the database. The final score $f(I)$ is then expressed by:

$$f(I) = Agg(S_{reg.}(R(i), I), ..., S_{reg.}(R(j), I))$$

Attention must be paid to the fact that each score $S_{reg.}(R(i), I)$ is an aggregated one (see 3): the score does not indicate if the set of regions in the image I is formed of the same number of regions, they just individually look like one of the regions $R(i)$.
As in the global case, the choice of Agg will change the request into a specific purpose. By setting Agg as the maximum, the user will get images containing at least one of the specified regions. With the minimum, he will get the images containing all of them. Again, he can assign weights by using an averaging operator. He can also specify regions he don't want to see in the retrieved images.

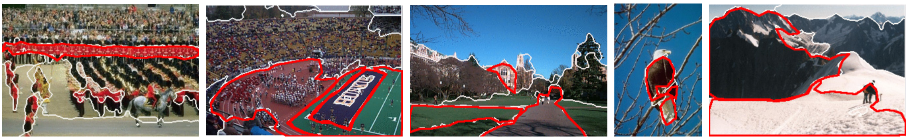

Fig. 3. Images used as requests to evaluate the efficiency of region similarity

Regions from different images: Obviously, the previous scheme can also be applied on a request composed of *regions extracted from different images.* $f(I)$ then takes the following form:

$$f(I) = Agg(S_{reg.}(R_1(i_1), I), ..., S_{reg.}(R_n(i_n), I))$$

In this case, the user compose a request from different sources. It is like he would build a squetch from different example regions.

The dynamic interface of our system is a practical tool to test all of these four schemes. With simple actions, an expert user can simulate one of these four requests, specify its parameters, and compose an aggregator to fulfill its needs. No current CBIR system offers all of these features. Though the similarity measures in our system are not fully developped yet, we have build a structure that let us implement, experiment and compare different methods of the image retrieval field.

5 Experimental Queries

As our system has been recently developed, we have not proceeded yet to its full evaluation. In this section, we only propose a preliminary essay based on five visual requests (see figure 3): each time, 2 or 3 regions of interest were selected, then used as a query. We choose to aggregate regional similarity measures (based on color, position and shape) using an averaging operator. The effectiveness of this aggregated measure is compared to the effectiveness of the global histogram similarity (one global histogram is extracted and compared to another histogram by a fuzzy similarity measure). The system retrieves the 80 best results for both methods among a database of 7700 images. This operation takes about 1.2 seconds. We evaluate the effectiveness of both methods by measuring two classical values:

- the rank RK of the first non pertinent image in the list.
- the proportion P of pertinent images in the first 10 results (extended to 20 or 30 when the first non-pertinent image was not present in the 10 first). The number of 10 results was chosen only to reflect the satisfaction of the first page of the results returned.

For a method to be efficient, it has to maximise the rank of the first non pertinent image, and also maximise the proportion of pertinent images in the first results.

On table 1, we observe that our method based on region matching gets better results than the global histogram comparison. Though the present preliminary evaluation shows encouraging results, a larger benchmarking should be runned to prove the effectiveness of our system.

Table 1. Evaluation Results

	Segmented		**Global Histo.**	
	P	RK	P	RK
image1	19/20	19	7/20	5
image2	22/30	19	24/30	21
image3	4/10	5	4/10	3
image4	8/10	6	4/10	2
image5	4/10	4	4/10	2
average results				
	0.65	10.6	0.47	6.6

6 Conclusion

Our image retrieval system uses a regional representation of images, built using a segmentation algorithm. This solution, currently investigated in the community, has been implemented in STRICT to let the user specify his visual interest in the request. For the expert, our system proposes features that enables him to modify parameters, build complex requests, and compare their results in a dynamic and easy to use interface.

We realize that the choice of the parameters and the operators used by the system to build complex queries may be difficult for the non-expert users. STRICT is in fact a CBIR experimentation platform that requires from users to know what they are doing, and for what purpose. A simpler version of the interface system is being developed to propose a ready-to-use image retrieval tool, including pre-chosen parameters and aggregation operators.

References

1. C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third Int. Conf. on Visual Information Systems*, Amsterdam, 1999.
2. Y. Chen and J.Z. Wang. A region-based fuzzy feature matching approach to content-based image indexing and retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9), September 2002.
3. H.D. Cheng, X.H. Jiang, Y. Sun, and J. Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34:2259–2281, 2001.
4. M. Detyniecki. *Mathematical Aggregation Operators and their Application to Video Querying*. PhD thesis, Universite Pierre et Marie Curie, 2000.

5. G. Durand and P. Faudemay. A fast region-merging segmentation algorithm for video analysis and indexing. In *CIR 98 Symposion and Workshop on Image Retrieval*, Newcastle, 1998.
6. J.P. Eakins. Towards intelligent image retrieval. *Pattern Recognition*, 35:3–14, 2002.
7. J. Li, J.Z. Wang, and G. Wiederhold. Irm: Integrated region matching for image retrieval. In *8th ACM Int. Conf. on Multimedia*, pages 147–156, 2000.
8. J.F. Omhover, M. Detyniecki, and B. Bouchon-Meunier. A region-similarity-based image retrieval system. In *IPMU'04*, Perugia, Italy, 2004 (submitted).
9. J.F. Omhover, M. Detyniecki, and M. Rifqi. Image retrieval using fuzzy similarity : Measure equivalence based on invariance in ranking. In *IEEE International Conference on Fuzzy Systems Fuzz-IEEE*, Budapest, Hungary, July 2004.
10. M. Rifqi, M. Detyniecki, and B. Bouchon-Meunier. Discrimination power of measures of resemblance. In *IFSA'03*, Istanbul, Turkey, 2003.
11. S. Santini and R. Jain. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):871–883, 1999.
12. M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.
13. A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.