IMAGE RETRIEVAL BASED ON TEXT AND VISUAL CONTENT USING NEURAL NETWORKS

Diego A. Castro[1], Leticia M. Seijas[1]

[1]Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires, Argentina
(dcastro,lseijas)@dc.uba.ar

Abstract

In the last few years there has been a dramatic increase in the amount of visual data to be searched and retrieved. Typically, images are described by their textual content (TBIR) or by their visual features (CBIR). However, these approaches still present many problems. The hybrid approach was recently introduced, combining both characteristics to improve the benefits of using text and visual content separately. In this work we examine the use of the Self Organizing Maps for content-based image indexing and retrieval. We propose a scoring function which eliminates irrelevant images from the results and we also introduce a SOM variant (ParBSOM) that reduces training and retrieval times. The application of these techniques to the hybrid approach improved computational results.

Key words: Image Retrieval, Self-Organizing Maps (SOM), Content-Based Image Retrieval (CBIR), Text-Based Image Retrieval (TBIR), ParBSOM, Scoring function

1 Introduction

In the last few years there has been a dramatic increase in the visual information available. Images generated from satellites, surveillance cameras, and even digital cameras produce a huge amount of information that gradually becomes more difficult to handle. In the image retrieval area (VIR), images are typically described by their textual content (TBIR) or by their visual features (CBIR). However, these approaches still present many problems. While in TBIR using natural language can lead to subjective and ambiguous descriptions, CBIR uses low-level features and can regard images as similar when they are semantically different -a problem known as semantic gap [4]. Recently, the hybrid approach was introduced. It combines both characteristics to improve the benefits of using text and visual content separately. CBIR nowadays is still far from being as well-matured as TBIR since it presents many challenges such as defining suitable descriptors and index structures. In this work we first focus on investigating techniques related to CBIR. We study one of the most popular image descriptors in the area: the color histograms [20]. We also investigate how Self-Organizing Maps (SOM) [7] can be used as an index in CBIR. SOM is an interesting alternative as it allows us to work with high-dimensional descriptors (typical case in CBIR).We propose a scoring function for images which eliminates irrelevant images from the results and we also introduce a SOM model that improves training and retrieval times (ParBSOM). In order to evaluate the performance of the studied methods, we base our experiments on image databases which are used in many works of the area or in events like ImageCLEF. Specifically, we use ZuBuD [17], UC-ID [16], UK Bench [13], and ImageCLEFphoto 2007 [5]. We also work with typical retrieval metrics such as Precision, Recall, F-Measure, and MAP [12]. In addition, we study how these techniques can be applied to the hybrid approach and provide computational results to assess their performance. Finally, we develop a research system known as Envision, which implements all the studied methods and was designed with extensibility and flexibility in mind. This paper is organized as follows: in Section 2 we explain the use of color histograms as image descriptors and present a scoring function to eliminate irrelevant images from the results. Section 3 describes SOM, variants and the proposed method ParBSOM focused on CBIR applications. Section 4 introduces the hybrid approach for VIR. In Section 5 standard image databases used throughout the experiments are described. Section 6 presents experimental results and finally in Section 7 we have concluding remarks.

2 Color Histograms

Color is one of the most intuitive features of an image which explains why color histograms [20] are among the most widely used features. The color histogram for an image is constructed by counting the number of pixels of each color.

One important aspect of color histograms is that they are invariant to rotation, mirroring, and scaling.

This descriptor can work with different color spaces such as RGB or HSV. In many works, HSV has been used as it is perceptually more uniform than the popular RGB [19] and the transformation from RGB can be performed in an easy and efficient way.

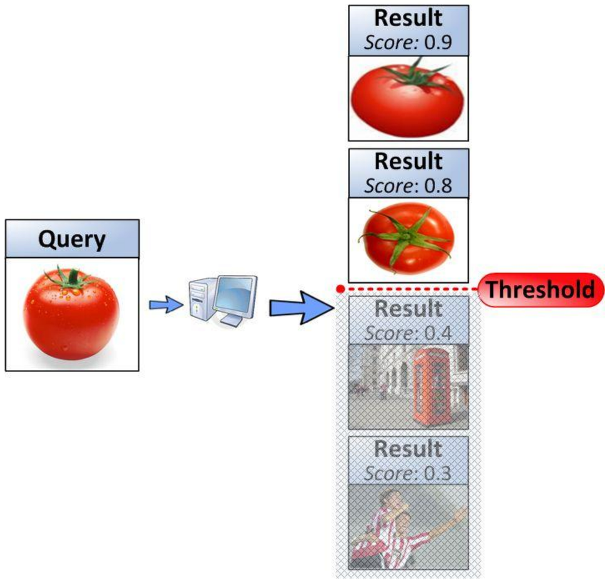

Figure 1. Threshold to eliminate irrelevant images during retrieval

To work with color histograms, a distance measure must be defined to determine how close images are. The L1 distance measure showed improved results in several works [6, 15].

It is also crucial to define how many bins will be used. Too many bins results in similar pictures being classified as different, while too few groups dissimilar ones together. In several works, 512 bins have been used [6].

In this work we used HSV color histograms with 512 bins and L1 distance. In Section 6 we provide computational results that support this decision.

Typically, in TBIR a scoring function is defined and used to retrieve only meaningful results. However, this topic has been neglected in CBIR. In order to eliminate irrelevant images from the results, we propose a scoring function that allows us to define a threshold (between 0 and 1) and filter those images below it (Fig.1). In Appendix A we formally demonstrate that this function is valid when using color histograms together with the L1 distance measure.

3 Content-based Image Retrieval using SOM

Content-based image retrieval is a problem that is getting more and more attention. The amount of visual data that has to be stored, managed, searched, and retrieved grows continuously on many fields of industry and research. Thus, CBIR is a challenging problem both in terms of effectiveness and efficiency.

One of the main problems faced in CBIR is that image descriptors are usually high-dimensional and current techniques such as R-Trees [1] or KD-Trees [1] are not scalable for dimensions higher than 20. In this context, SOM is an interesting alternative as it allows us to work with high-dimensional descriptors. SOM acts as an image classifier, mapping images to neurons in the network. It generates maps where similar images are close in the network and these characteristics are used during retrieval.

When manipulating huge databases, a good index is a necessity. Processing every single item in a database when doing queries is extremely inefficient and slow. When working with text-based documents, creating good indices is not very difficult. Simply maintaining a list of all words in the database and information on which documents contain which words is enough. When searching for images, however, this approach gets more complicated. Raw image data is non-indexable as such, so the feature vectors must be used as the basis of the index. The problem we now face is that indexing data points in a multidimensional vector space is a non-trivial task.

We examine the use of the Self-Organizing Maps (SOM) as a tool for content-based image indexing and retrieval.

3.1 SOM and variants

3.1.1 SOM

Self-Organizing Maps are unsupervised neural networks that provide mapping from high-dimensional input space to a usually two-dimensional regular grid while preserving topological relations as faithfully as possible [7].

The SOM consists of a set of i units or nodes arranged in a two-dimensional grid, with a weight vector $w_i \in \Re^m$attached to each unit.

$$\begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{bmatrix}$$

$$w_i = [v_{i1}, \dots, v_{im}]$$

Elements from the high-dimensional input space, referred to as input vector $x \in \Re^m$, are presented to the SOM and the activation of each unit for the presented input vector is calculated using an activation function. Then, the *Best Matching Unit* (BMU) is selected as the node associated with the weight vector $w_*$with the smallest distance (Equation 1).

$$dist(x, w_*) = \min_{\mathrm{i}=1\dots\mathrm{n}} (dist(x, w_i)) \quad (1)$$

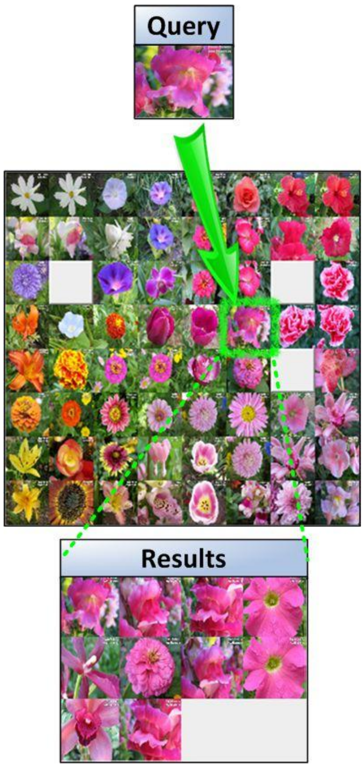

Figure 2. Retrieving images from a trained SOM. Color histogram was used as image descriptor.

The next step consists of performing a reduction of the difference between the input vector and the weight vector of the BMU w_*, moving it to the presented input vector by a certain fraction of the distance as indicated by a time-decreasing learning rate α. The weight vectors of units in the neighborhood around the BMU as described by a time-decreasing neighborhood function $h_{i,*}$ are modified accordingly, although to a less strong amount as compared to the BMU. The learning rule of the algorithm is defined as

$$w_i(t+1) \coloneqq w_i(t) + \alpha(t)h_{i,*}(t)(x(t) - w_i(t)) \tag{2}$$

where t denotes the current learning iteration, α represents the time-varying learning rate, $h_{i,*}$ is the time-varying neighborhood function, x is the current input vector and w_i is the weight vector assigned to unit i. This learning procedure finally leads to a topologically ordered mapping of the presented input vectors. Thus, similar input data is mapped onto neighboring regions on the map. The map is called a *topological feature map*, and preserves the similarity of the input data in feature spaces clustering mutually similar feature vectors in neighboring nodes (see Fig.2).

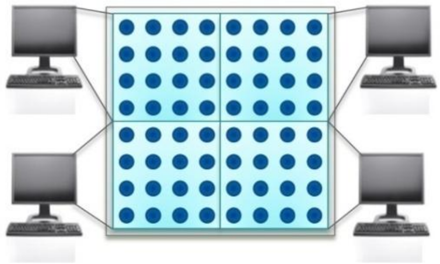

Figure 3. In ParSOM the SOM is divided into multiple regions, each one assigned to a processing node, allowing parallel execution.

Despite these advantages, SOM applications have been limited. The search for the BMU dominates the computing time of the SOM algorithm, making it computationally expensive for high-dimensional inputs or large SOM networks. To overcome these difficulties several models have been developed [9, 10, 8, 21, 14].

3.1.2 ParSOM

The ParSOM [21] is a software-based parallel implementation of the Self Organizing Map, using a simple asymmetric model of parallelization in order to speed up time during data analysis, for example, in training and retrieval. A serial implementation of the SOM iterates through the training algorithm. This involves scanning all units to find the BMU and then modifying all units to reflect the change. The main idea of ParSOM is based on partitioning, with one master controlling multiple slaves. By segmenting the SOM into multiple regions, the memory usage can be effectively distributed across the processing nodes (see Fig.3). A master thread selects an input vector and broadcasts it to all its slave threads. All threads search for BMUs, concurrently. These best matches are sent back to the master, which determines a global match. All slaves are notified of the global match and they modify their units relative to the location of it independently. This parallel approach allows reducing training and retrieval times for CBIR using large databases while maintaining the benefits of the topology preserving capability of the SOM.

3.1.3 BSOM

Batch SOM is a variant of traditional SOM, developed by Teuvo Kohonen in 1990 [8]. The Batch version consists of modifying the procedure the weight vectors of the map are adapted. Instead of modifying the weights each time an input pattern is presented, as in the traditional algorithm described in Section 3.1.1, BMUs are calculated for all input vectors in the training set and then the network is updated. BSOM constitutes a deterministic approach [2]. The learning rule is defined as

$$w_i \coloneqq \frac{\sum_{j=1}^{N} h_{i,*_j}(t) x_j}{\sum_{j=1}^{N} h_{i,*_j}(t)} \tag{3}$$

where *i* refers to a specific unit of the map, *t* denotes the current learning iteration, *N* is the number of training patterns, *h* represents the neighborhood function, and $*_j$ is the BMU associated with pattern x_j. Note that in the Batch version no learning parameter has to be defined.

In terms of time-complexity BSOM improves efficiency with respect to traditional stochastic SOM, when the difference between the number of input patterns *(N)* and the number of units in the map *(n)* increases, being typically $N >> n$.

3.2 A proposed variant for indexing: ParBSOM

We propose the use of Parallel Batch SOM variant (ParBSOM) based on the idea of combining the remarkable properties of ParSOM and BSOM described in Sections 3.1.2 and 3.1.3 respectively.

The objective is to obtain a reduction in training and retrieval times maintaining the desirable properties of BSOM for CBIR, without reducing the quality of the final map.

As in the case of ParSOM, ParBSOM uses multiple processors executing in parallel, each one dedicated to the search of the BMU and the adaptation of weight vectors belonging to a defined region of the map. This parallel proposal is implemented for the Batch version of the SOM, so all the desirable characteristics that are valid for BSOM, remain in ParBSOM.

In general, most parallel implementations are either network-partition or data-partition based. In the first, the parallelization is achieved by segmenting the original map into smaller sub maps. On the other hand, data-partitioning techniques break down the set of input data into smaller disjoint sets along the processing nodes, each node training a complete copy of the map. Previous works have mentioned parallel implementation of BSOM [11, 18] mainly oriented to Data-Mining applications. Focused on the necessity of sorting through large amounts of data and picking out relevant information, several approaches have been proposed. In [11] three variants of parallelization are described: Network-partitioning SOM based on the stochastic SOM algorithm, Data-partitioned BSOM, and Data-partitioned sparse BSOM oriented to datasets often containing a large fraction of zero entries. The authors present experiments based on maps containing 16 and 64 neurons, using several training sets. As an example, we mention one of them containing 99,984 patterns with an associated dimension of 272. The data-partitioned parallel method based on the batch SOM algorithm has been considered a computational efficient approach for this kind of application.

In our case, we consider the network-partitioning implementation based on the BSOM well-suited as an index for CBIR. The experiments (see Section 6) were carried out on maps and training sets of different sizes. For example, a map containing 1,000 neurons was trained using a data set composed by 10,000 patterns with an associated dimension of 500. Important improvements were obtained using only two processing nodes.

4 The Hybrid Approach

In order to overcome TBIR and CBIR problems, the hybrid approach was recently introduced. Various studies have shown that the combination of contentbased and text-based approaches can lead to better results than using both approaches separately [22, 3, 4].

There are several possibilities to achieve such a combination of TBIR and CBIR results. For example, linear expansion model [22] treats TBIR and CBIR equally, that is, items with same value of their contribution in TBIR and CBIR are handled with the same importance. While text-based retrieval is a well-matured field of research and can therefore be a very efficient procedure for image retrieval, CBIR often produces many irrelevant results because of the comprehensive semantics of images. In other words, even though an image is much similar to the sample image in visual features, it may have a far distance in semantics. To overcome this limitation, the refinement model [4] to integrate CBIR into TBIR was introduced. It reorders TBIR results using the results of CBIR. This strategy gives more importance to textual results as nowadays TBIR is a much more advanced area than CBIR.

Fig.4 shows the typical structure of a hybrid system. During the index construction stage, each module works separately (offline). During the retrieval stage (online), queries could be formulated using textual information, images or both. In the last case, each module produces its own results and then both lists are merged (*late fusion*).

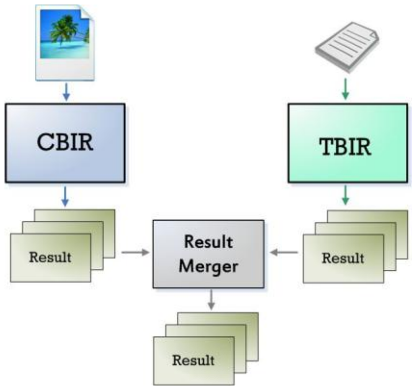

Figure 4. Typical organization of a hybrid system

5 Performance Evaluation in VIR

Over the last few years VIR has become a significant area of research and a huge number of systems have been developed. Unfortunately, as many authors use their own image sets and retrieval metrics, comparing the quality of VIR systems has become virtually impossible.

In this section we describe standard image databases that will be used throughout the experiments.

5.1 Databases

Currently, there are very few image databases available that can be used for evaluating VIR systems. In the following section, we will present a list of open and free databases. Detailed information of each database can be found in Table 1.

Table 1: Detailed information of image databases

	ZuBuD	UCID	UK Bench	ImageCLEF 2007
Type	CBIR	CBIR	CBIR	TBIR + CBIR
Images	1,005	1,338	10,200	20,000
Queries	105	262	2,550	60
Images per query	5	2.5 (average)	3	57 (average)
Image format	PNG	TIF	JPG	JPG

Figure 5. Different images of the same building (ZuBuD)

Figure 6. Example images from UCID database

5.1.1 ZuBuD

ZuBuD (*Zurich Buildings Database* [17]) has been created by the *Swiss Federal Institute of Technology*. This database contains pictures of buildings from Zurich, some of them taken from different angles or under different weather conditions.

ZuBuD consists of two parts: a training part of 1,005 images of 201 buildings and a query part of 115 images. Each query contains one of the buildings from the main part of the database. Given a query image, only images showing exactly the same building are considered relevant (Fig.5). This database can be used to evaluate CBIR systems.

5.1.2 UCID

UCID (*Uncompressed Colour Image Database* [16]) has been created by the *Nottingham University*. It provides a standard image set to compare CBIR systems as well as image compression applications.

This database consists of images from different topics and relevance assessments that were created manually. Example images can be found in Fig.6.

Figure 7: Different objects from UK Bench database

5.1.3 UK Bench

UK Bench [13] has been developed for object recognition tasks but can also be used to evaluate CBIR systems. It has been created by the *University of Kentucky* and contains different type of objects, such as cd covers, flowers and toys. Each object has 4 images taken from different angles and under different conditions. Fig.7 shows some pictures from the database.

5.1.4 ImageCLEFphoto 2007

During the ImageCLEF competition organized in 2007, the *IAPR TC-12 Benchmark* [5] database has been used for the photo retrieval task. This database contains tourist and sport photographs with semi-structured multilingual captions (in this work we only use English captions). An example is shown in Fig.8.

6 Experimental Results

In this section results related to retrieval performance and quality of the studied methods are presented.

First, we performed several experiments in order to determine how many bins and which distance metric should be used with HSV color histograms. We compared the performance in all databases using different number of bins (64, 128, 256, and 512) and different distance metrics (L1 and L2). We also used a fixed number (k) of results for each query performed. Table 2 and Fig.9 show results for ZuBuD using k=20. Results using the other databases and different values of *k* follow the same pattern. It can be seen that L1 outperforms L2 and that using 512 bins leads to a quality improvement.

```
<DOC>
<DOCNO>annotations/16/16019.eng</DOCNO>
<TITLE>Flamingo Beach</TITLE>
<DESCRIPTION>a photo of a brown sandy beach; the dark
  blue sea with small breaking waves behind it; a dark
  green palm tree in the foreground on the left; a blue
  sky with clouds on the horizon in the background;
</DESCRIPTION>
<NOTES>Original name in Portuguese: "Praia do Flamengo";
  Flamingo Beach is considered as one of the most
  beautiful beaches of Brazil;</NOTES>
<LOCATION>Salvador, Brazil</LOCATION>
<DATE>2 October 2004</DATE>
<IMAGE>images/16/16019.jpg</IMAGE>
<THUMBNAIL>thumbnails/16/16019.jpg</THUMBNAIL>
</DOC>
```

Figure 8: Image and text from ImageCLEFphoto 2007 database

Table 2: Performance evaluation in ZuBuD using different number of bins and distance metrics (k=20)

	MAP	Prec.	Recall	F
L1 - 64 bins	72.71	20.91	83.65	33.46
L2 - 64 bins	63.46	19.30	77.22	30.89
L1 - 128 bins	77.69	21.35	85.39	34.16
L2 - 128 bins	66.81	19.56	78.26	31.30
L1 - 256 bins	81.08	22.09	88.35	35.34
L2 - 256 bins	68.10	19.61	78.43	31.37
L1 - 512 bins	**83.42**	**22.52**	**90.09**	**36.03**
L2 - 512 bins	66.88	19.09	76.34	30.54

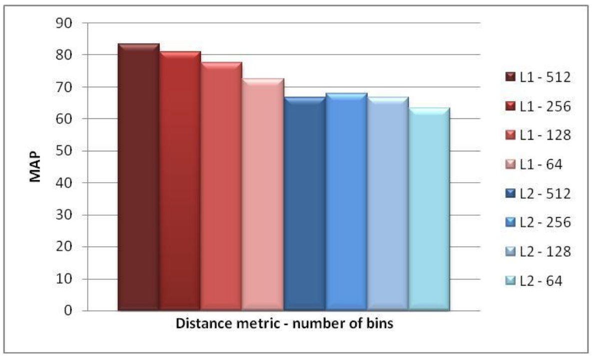

Figure 9: Evaluation of MAP in ZuBuD using different number of bins and distance metrics (k=20)

Table 3: Improvements in training times (10 epochs of training)

Data [size x dimension]	**Map units**	BSOM vs. SOM	ParSOM vs. SOM	ParBSOM vs. BSOM	ParBSOM vs. ParSOM
[5.000 x 250]	500 1.000	56% 52%	33% 33%	**34%** **35%**	**57%** **54%**
[5.000 x 500]	500 1.000	55% 50%	31% 31%	**34%** **40%**	**58%** **57%**
[10.000 x 250]	500 1.000	60% 58%	32% 32%	**36%** **37%**	**63%** **61%**
[10.000 x 500]	500 1.000	59% 56%	31% 32%	**38%** **40%**	**63%** **61%**

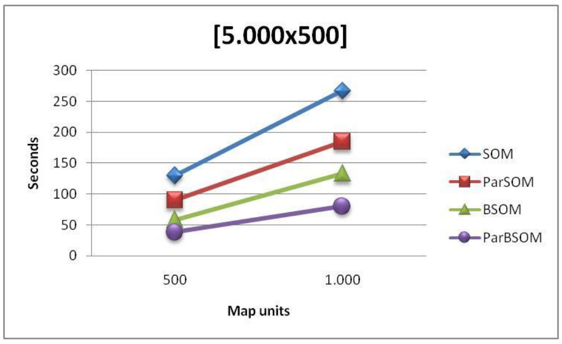

Figure 10: Improvements in training times (10 epochs of training and 5,000 patterns of 500 dimensions)

In Table 3 and Fig.10, we compared training times for different SOM models: the traditional SOM, BSOM, ParSOM, and our proposed variant ParBSOM. Data sets of different size and dimension and two processing nodes –for parallel versions- were used in the experiments. As expected, the existing variants (BSOM and ParSOM) reduce training times (above 50% and 30% respectively). In addition, our proposed method improves BSOM by about 40% and also ParSOM by about 60%.

Using the databases described in Section 5, we focused on measuring the quality of the generated maps. We compared ParSOM and ParBSOM with the *Brute Force* algorithm, which consists of performing a linear search through the database. Table 4 and Fig.11 show that ParBSOM loses less than 10% of quality in all databases against the Brute Force method and that ParSOM has a similar behavior.

Table 4: Quality loss in terms of F-Measure

Image DBs	**Quality loss** ParBSOM vs. Brute Force	**Quality loss** ParBSOM vs. Brute Force	**Quality loss** ParBSOM vs. ParBSOM
ZuBuD	**0.46%**	1.12%	0.66%
UCID	**8.1%**	10.89%	3.04%
UK Bench	**9.07%**	9.94%	0.97%

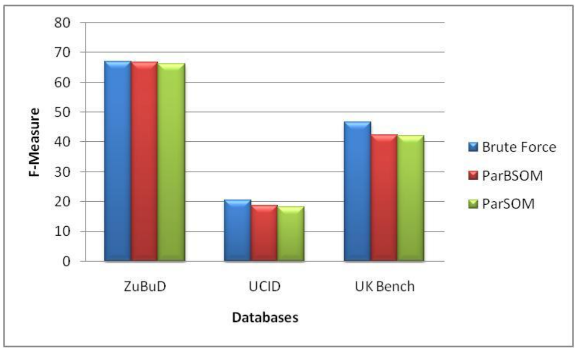

Figure 11: Quality loss in terms of F-Measure

In spite of losing some quality, ParBSOM considerably improves retrieval times (more than 90%) compared to the Brute Force algorithm, as can be observed in Table 5 and Fig.12.

Table 5: Time required to retrieve an image from the database

Image DBs	**Time** Brute Force	**Time** ParBSOM	**Improvement** ParBSOM vs. Brute Force
ZuBuD	3.43 ms	0.27 ms.	**92%**
UCID	4.58 ms.	0.32 ms.	**93%**
UK Bench	40.63 ms.	1.68 ms.	**96%**

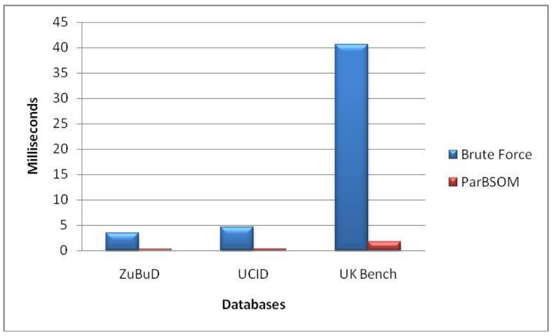

Figure 12: Time required to retrieve an image from the database

Finally, we applied the studied methods (HSV color histograms and ParBSOM) to a hybrid system which uses the refinement strategy (Table 6 and Fig.13). As expected, metrics which are not sensitive to image rankings (Precision, Recall, and F-Measure) show no changes as refinement alters TBIR rankings without modifying the results set. MAP and Precision in the first 10 and 20 results show an improvement between 10% and 20%.

Table 6: Different retrieval methods for ImageCLEFphoto 2007

Metric	**TBIR**	**Hybrid**	Improvement
MAP	14.94	16.59	9.95%
Precision	5.35	5.35	0%
Recall	49.27	49.27	0%
F-Measure	8.26	8.26	0%
Prec(10)	22.33	27.83	**19.76%**
Prec(20)	18.33	22.08	**16.98%**

7 Conclusions

We have studied several techniques applied to VIR. First, we focused on color histograms, comparing their performance in the RGB and HSV space. We have proposed a scoring function for color histograms in order to eliminate irrelevant images from the results.

Then, we have investigated how SOM can be used as an index in CBIR.We have introduced a SOM variant (ParBSOM) that improves BSOM’s training time by about 40% and also ParSOM’s training time by about 60%.

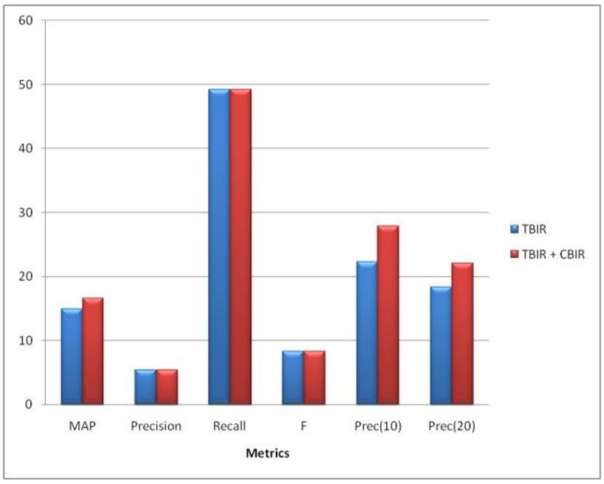

Figure 13: Different retrieval methods for ImageCLEFphoto 2007

We have studied hybrid techniques and observed that the refinement strategy can actually improve textual results by using visual features. The application of the proposed scoring function and the ParBSOM method to the hybrid approach improved computational results related to the retrieved data quality besides reducing training and retrieval time.

Despite the fact that VIR has been one of the most active research areas, there are many open issues that still need to be addressed. In the future we intend to investigate image descriptors that combine color with other interesting features such as shape or texture. We also plan to focus on developing new hybrid techniques to combine textual and visual results.

Appendix A

In this section we show how the scoring function of Section 2 can be defined. We start with showing that the L1 distance between two color histograms can be normalized. Let H, H0 be color histograms (with n bins each), we want to define the following function

$$normalize\left(d_{L1}(H,H')\right) \in [0,1] \tag{4}$$

By color histogram definition, we know

$$\sum_{i=1}^{n} H[i] = \sum_{i=1}^{n} H'[i] = 1 \tag{5}$$

$$\forall i = 1 \ldots n \;\; H[i] \geq 0 \;\; y \;\;\; H'[i] \geq 0 \tag{6}$$

Then, by L1 norm definition

$$d_{L1}(H, H') = \sum_{i=1}^{n} |H[i] - H'^{[i]}| \tag{7}$$

Assuming that a and b are both greater than 0, we know that the following properties hold

$$|a + b| = a + b \tag{8}$$

$$|a - b| \leq a + b \tag{9}$$

Then,

$$\begin{aligned} d_{L1}(H, H') \quad &= \sum_{i=1}^{n} |H[i] - H'[i]| && \text{(by definition)} \\ &\leq \sum_{i=1}^{n} |H[i] + H'[i]| && \text{(by Equation (9))} \\ &= \sum_{i=1}^{n} H[i] + \sum_{i=1}^{n} H'[i] && \text{(by Equation (8))} \\ &=1+1 && \text{(by Equation (5))} \\ &=2 \end{aligned}$$

Now, using that dL1(H,H0) _ 2, we define normalize function as:

$$normalize\big(d_{L1}(H, H')\big) = \frac{d_{L1}(H, H')}{2} \in [0,1] \tag{10}$$

Using Equation (10), we can define the scoring function as

$$score(H, H') = 1 - normalize\big(d_{L1}(H, H')\big) \tag{11}$$

References

1. Böhm C., Berchtold S., Keim D., 2001, *Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases*, ACM Comput. Surv., 33, 3, pp. 322–373.
2. Fort J. C., Cottrell M., Letremy P.. 2001, *Stochastic on-line algorithm versus batch algorithm for quantization and self organizing maps*, Neural Networks for Signal Processing XI: Proceedings of the 2001 IEEE Signal Processing Society Workshop, pp. 43–52.
3. Gong Z., Liu Q., Zhang J., 2006, *Web Image Retrieval Refinement by Visual Contents*,In: J.X. Yu, M. Kitsuregawa, and H.V. Leong (Eds.), WAIM 2006, LNCS 4016, Springer-Verlag Berlin Heidelberg, pp. 143–145.
4. Grubinger M., 2007, *Analysis and Evaluation of Visual Information Systems Performance*, PhD Thesis, School of Computer Science and Mathematics, Faculty of Health, Engineering and Science, Victoria University, Melbourne, Australia.
5. Grubinger M., Clough P., Hanbury A., M¨uller H., *2008, Overview of the ImageCLEFphoto 2007 Photographic Retrieval Task*, Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Revised Selected Papers, Springer-Verlag, pp. 433–444.
6. Jonsg°ard O., 2005, *Improvements on colour histogram-based CBIR*, PhD Thesis, Gjøvik University College, Stockholm, Norway.
7. Kohonen T., 1982, *Self-organized formation of topologically correct feature maps*, Biological Cybernetics, 43, pp. 59–69.
8. Kohonen T., 2001, *Self-Organizing Maps*, Springer-Verlag.
9. Koikkalainen P., Oja E., 1990, *Self-organizing hierarchical feature maps, IJCNN International Joint Conference on Neural Networks*, 2, pp. 279–284.
10. Laaksonen J., Koskela M., Laakso S., Oja E., 2000, *Picsom - Content-based image retrieval with self-organizing maps*, Pattern Recognition Letters, 21, 13–14, pp. 1199–1207.
11. Lawrence R. D., Almasi G. S., Rushmeier H. E., 1999, *A Scalable Parallel Algorithm for Self-Organizing Maps with Applications to Sparse Data Mining Problems,* Data Mining and Knowledge Discovery, 3, pp. 171–195.
12. Manning C., Raghavan P., Sch¨utze H., 2009, *An Introduction to Information Retrieval*, Cambridge University Press.
13. Nist´er D., Stew´enius H., 2006, *Scalable Recognition with a Vocabulary Tree, IEEE* Conference on Computer Vision and Pattern Recognition (CVPR), Washington, 2, pp. 2161–2168.
14. Rauber A., Merkl D., Dittenbach M., 2002, *The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data*, IEEE Transactions on Neural Networks, 13, pp. 1331–1341.

15. Russell R., Sinha P.,2001, *Perceptually-based Comparison of Image Similarity Metrics*, MIT AIM 2001-014.
16. Schaefer G., Stich M., 2004, UCID - *An Uncompressed Colour Image Database, Storage and Retrieval Methods and Applications for Multimedia* 2004 (Proceedings of SPIE), San Jos´e, 5307, pp. 472–480.
17. Shao H., Svoboda T., van Gool L., 2003, *ZuBuD — Zurich Buildings Database for Image Based Recognition*, Techn. Report 260, Swiss Federal Institute of Technology, Switzerland.
18. Silva B., Marques N., 2007, *A Hybrid Parallel SOM Algorithm for Large Maps in Data-Mining,* New Trends in Artificial Intelligence, APPIA.
19. Smith J., Chang S., 1995, *Single color extraction and image query,*ICIP ’95: Proceedings of the 1995 International Conference on Image Processing,IEEE Computer Society, Washington, 3, 3528.
20. Swain M., Ballard D., 1991, *Color indexing*,International Journal of Computer Vision, 7, pp. 11–32.
21. Tomsich P., Rauber A., Merkl D., 2000, parSOM: *Using parallelism to overcome memory latency in self-organizing neural networks*, In: High Performance Computing and Networking, Society Press, pp. 61–5.
22. Zhuang Y., Qing L., RynsonW., 2001, *Web-Based Image Retrieval*: A Hybrid Approach, Computer Graphics International, pp. 62–72.