Region Based Image Similarity Search Inspired by Text Search *

Giuseppe Amato
ISTI-CNR
Pisa, Italy
Giuseppe.Amato@isti.cnr.it

Vanessa Magionami
ISTI-CNR
Pisa, Italy
Vanessa.Magionami@isti.cnr.it

Pasquale Savino
ISTI-CNR
Pisa, Italy
Pasquale.Savino@isti.cnr.it

Abstract

We present a novel technique for processing image similarity search by using an approach that takes inspiration from text retrieval techniques. In our approach images are indexed by using visual terms taken from a visual lexicon obtained clustering regions of images in the dataset. A weighting and matching schema is defined that allow effective image retrieval to be performed by using inverted files, thus requiring reduced storage space and achieving high efficiency.

1 Introduction

Retrieval of visual documents, in multimedia digital libraries, requires new search paradigms. Recently similarity search or content based retrieval was proposed as a viable alternative. In this paper, we present an approach to content-based image retrieval that takes inspiration from text retrieval techniques [5], as in [1, 6, 2]: images are indexed and retrieved by means of *visual terms*. A visual term is a prototype representing a typical visual region that can be found in an image. The key component of our approach is the *visual lexicon*. The visual lexicon is a set of prototypes of visual regions, that is a set of typical regions, which can be found in an image. We call *visual term* a region prototype. An image is indexed by associating it with the set of visual terms that are judged to be contained in it, similarly to a text document that is indexed by the set of terms that are contained in it. There are some issues that have to be addressed to obtain that.

1. *Visual lexicon generation*: how the set of visual terms is chosen and how are terms represented?
2. *Image indexing*: which visual terms are associated with an image?
3. *Image retrieval*: how image queries are matched against indexed images?

*This work was partially supported by the DELOS NoE and the Multimatch project, funded by the European Commission under FP6 (Sixth Framework Programme).

2 Visual lexicon generation

In text retrieval systems there are well defined morphological rules to decide which are the terms of the lexicon. However, in our context it is not obvious what is the nature of a visual term. Regions contained in images might have infinite variations and using all possible regions has no sense. However, physically different regions can play the same role from a visual point of view. Intuitively, regions that play the same role have to be represented by the same visual term. We build the visual lexicon by choosing a training set of images and by applying a segmentation algorithm to extract all possible regions. We use a clustering algorithm to group together visually similar regions. The representative of each obtained cluster is a visual term.

Let us describe this process more precisely. Visual similarity of regions is judged by extracting low level features from them and by comparing the extracted features by means of a similarity (or dissimilarity) function. There are several low lever features that can be extracted from regions, which consider various aspects to judge the visual similarity. Choosing different features may lead to different results of the clustering algorithms. For instance, the use of color histograms groups together regions having similar colors, while the use of shape descriptors clearly groups together regions having the same shape. We propose to group together regions according to various features so that we obtain a multi-feature visual lexicon containing visual terms obtained by considering different features. The multi-feature lexicon is the union of a number of mono-feature visual lexicons. For instance, there will be visual terms that represent group of regions having similar shapes and terms that represent regions having similar colors. Specifically, the clustering algorithm is applied multiple times to the training set. In each application a different visual descriptor is used. Each application returns a mono-feature visual lexicon consisting of a set of prototypes representing

regions that are judged to be similar, according to one visual descriptor.

In our experiments, we have used the ITI Segmentation algorithm [3] to segment the images. We used the five MPEG-7 visual descriptors [4] to represent the features extracted from regions. Finally we have used the simple k-means algorithm to cluster regions.

3 Image indexing

Once we have a visual lexicon, we have to define how visual terms are associated with images. In principle, a visual term should be associated with an image when the image contains a region represented by the visual term. To do that, every image is first segmented into regions and the low level features, the same used for building the visual lexicon, are extracted from them. The terms are chosen by selecting, for each extracted region, the n most similar visual terms in each mono-feature visual lexicon, where n is a parameter that can be tuned to optimize the performance of the approach. Suppose the multi-feature visual lexicon is composed of f mono-feature lexicons, each region is associated with $n \cdot f$ visual terms. The set of terms corresponding to a region can be considered as different *senses* of the region.

To consider the relevance of a visual term in an image we also give a weight to the selected visual terms. The weighting strategy that we use is inspired by the $TF * IDF$ technique typically used in text indexing systems. TF gives the importance of a terms in the document (its frequency in text retrieval systems). IDF is the importance of a term with respect to the entire dataset (the inverse of the frequency in the dataset). We use the same terminology and we re-define them according to our context. The weight w_t^I of term t in image I is $w_t^I = TF_t^I * IDF_t$.

Intuitively, the TF_t^I of term t in image I should be directly proportional to similarity between a region and the visual term and to the size of the region in the image. In addition, it should be directly proportional to the number of regions, in the image, represented by t. This can be expressed as $TF_t^I = \sum_{r \in Regions(I,t)} sim(r,t) * cover(r,I)$, where $Regions(I,t)$ is the set of regions of I that are represented by t, $sim(r,t)$ gives the similarity of region r to the visual term t, according to the low level feature of the lexicon of t, and $cover(r,I)$ is the percentage of the area covered by r in I.

The IDF_t is defined as in traditional text retrieval systems. It is the logarithm of the ratio between the dataset size N and the number n_t of images, which t is associated with: $IDF_t = \log_e \frac{N}{n_t}$.

Note that this indexing schema has the effect to select the most relevant terms for an image in a given collection. Given that terms are obtained by using different low level features, an implicit outcome of this approach is that we are also able to automatically select and combine the most relevant features for a given image in a given collection.

4 Image retrieval

Suppose we have a query image I_q. We want to search for the k most similar images to I_q. This is obtained by indexing the image I_q as described in previous section. The similarity between an image I of the dataset and the query image I_q is obtained by matching the terms respectively associated. More specifically, we use the vector space model for doing that. Every image I is supposed to be associated with a vector of weights W_I. The vector contains an element for each visual term of the visual lexicon containing the weight of the term for the image. The weight of the term is 0 if the term is not associated with the image, it is computed as discussed in previous section elsewhere.

Similarity between two images I_1, I_2 is computed as the cosine between their vectors of weights, which in case of normalized vectors can be computed as the dot product between the two vectors.

This indexing schema allows using inverted files, which are widely used for efficient text retrieval, to perform image retrieval. This, in addition to obvious efficiency advantage with respect to other access structures, has also the advantage to save memory space. In fact, typically all features extracted by all images have to be stored somewhere to be matched against the query. In this case just the representation of the visual terms in the lexicon should be stored, along with the weights of the regions in the images with a significant reduction of required storage space.

5 Experiments

We have carried-out comprehensive experiments to investigate the system effectiveness and efficiency. Here we briefly report the comparisons against the use of pure MPEG-7 [4], the SIMPLIcity system [6], and KeyBlock [2]. For the comparison with the Simplicity system we have re-implemented its indexing and retrieval schema using the segmentation and feature extraction tools we have used. For the comparison with KeyBlock we have adopted their segmentation and feature extraction techniques in our system.

5.1 Comparison with MPEG-7

In this test we compared our approach, according to various settings, with the direct use of the MPEG-7 descriptors [4]. MPEG-7 offers five descriptors which take into considerations different visual aspects in an image. Each descriptor is associated with a similarity function which can be used to judge the similarity between two images according

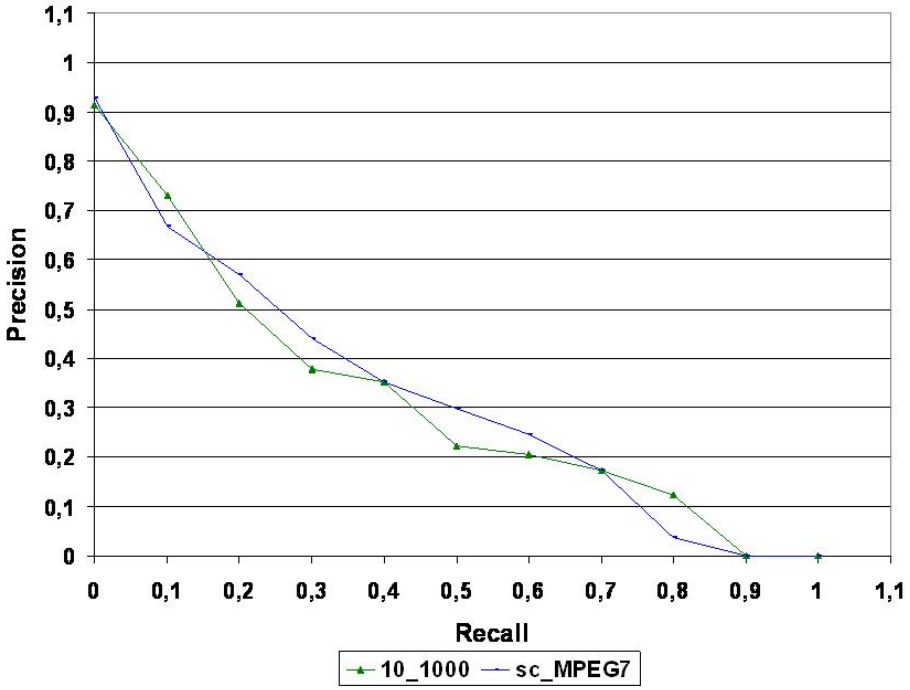


Figure 1. Comparing our technique with use of pure MPEG-7. In our method we used a lexicon of size 1000 and each region was associated with 10 visual terms. For the MPEG-7 method we used the Scalable Color descriptor, which resulted to be the best among all descriptor, in the dataset that we used. Performance of our approach is comparable to the best MPEG-7 descriptor.

to a specific descriptor. Image retrieval can be performed by extracting a descriptor from a query image and by searching the k most similar images according to the chosen descriptor. Our objective here is to compare our proposal with the direct use of the specific MPEG-7 descriptors.

5.1.1 Description of the experiments

For this test we used a collection of 10000 images (a subset of the Department of Water Resources in California Collection) stored in JPEG format with size 384X256 or 256X384 containing scenes of various types. Our approach was tested using various configurations according to various sizes of the lexicon, number of senses assigned to each region (see Section 3), and descriptors used. We tested separately the various descriptors, and we also combined all descriptors together, having our indexer determine the importance of the various descriptors in the various images.

The direct use of MPEG-7 was tested using all MPEG-7 descriptors independently. Different descriptor might give different results according to different queries.

We used a TREC-like approach for executing the tests. Union of results obtained by the various configurations of our approach and by the direct use of MPEG-7 were ranked by a user and used to judge the performance of the various systems.

5.1.2 Experiment settings

The entire dataset was segmented using the ITI segmentation algorithm [3]. The ITI algorithm was set to extract at most 10 regions from each image. From each region we extracted the five MPEG-7 visual descriptors (Scalable Color, Edge Histogram, Dominant Color, Region Shape, Homogenous Texture), by using the MPEG-7 reference software.

The regions belonging to a subset of 1000 images were used as the training set given as input to the clustering algorithm for the generation of the visual lexicon (see section 2).

Two different visual lexicons were generated. The first visual lexicon contains 100 visual terms, the second one contains 1000 visual terms.

(a) Africa people and villages (b) Beach (c) Buildings (d) Buses (e) Dinosaurs

(f) Elephants (g) Flowers (h) Horses (i) Mountains and glaciers (j) Food

Figure 2. Examples of images for each category

Both lexicons were separately used to index the entire dataset. More specifically, each lexicon was used to index the entire dataset multiple times according to different indexing parameters. The visual lexicon of size 100 was used to independently index the dataset using 1, 5, and 10 senses (see section 3) for each region. The visual lexicon of size 1000 was used to independently index the dataset using 1, 5, 50 , and 100 senses for each region.

Our approach was tested by individually using the mono-feature visual lexicons and by combining the mono-feature visual lexicons in a single multi-feature lexicon (combined method).

We used 15 different queries to perform the experiments. For each query, the union of the first 40 retrieved images returned by each configuration and by the pure MPEG-7 retrieval techniques is considered to be the candidate set of relevant documents. This set of documents is ranked by a user according to the relevance to the query. The obtained ranking is used to compare the various configurations of our approach and the different MPEG-7 based techniques. Precision and recall was used as a performance measure.

5.1.3 Results

We have observed that the combined method (the use of a multi-feature lexicon) offered better performance both in case of lexicons of size 100 and 1000. In case of lexicon of size 100 the best performance was obtained using 5 senses per regions. In case of lexicons of size 1000, the best performance was obtained with 10 senses per region.

For what concerns the direct use of the MPEG-7 descriptors the best performance was obtained with the scalable color descriptor.

Figure 1 shows the comparison between our combined method and the direct use of the scalable color MPEG-7 descriptor. Our method results to be almost equivalent to it. The advantage in our case is that we do not have to choose in advance the correct descriptor (scalable color in this case is the best), given that our indexing method automatically adapts the weight of the various components. In fact, the direct use of the other MPEG-7 descriptor present a performance that is much worse than the scalable color descriptor.

5.2 Comparison with Simplicity

Simplicity [6] is a system that also uses region based retrieval. Our objective here is to compare Simplicity and our approach in terms of the pure indexing and retrieval functionality.

5.2.1 Description of the experiments

We have performed two different tests. The first is very similar to the comparison with the direct use of MPEG-7 descriptors. We have used also in this case a TREC like approach to compare the two systems. In this case the database that we have used is the COREL database.

The second test is exactly the same test executed in the Simplicity paper [6]. In this test a subset of the COREL collection with images classified in different classes was used. We checked how the two systems were able to retrieve images of specific classes.

5.2.2 Experiment settings

In order to perform an objective comparison we have re-implemented the Simplicity indexing and retrieval algorithms, however differently from the original Simplicity system, we have used also in our Simplicity implementation the ITI[3] segmentation tool and the MPEG-7 reference software to respectively extract region and describe them in terms of visual features.

For the first experiment we have used the entire COREL collection, consisting of about 60000 images. All images

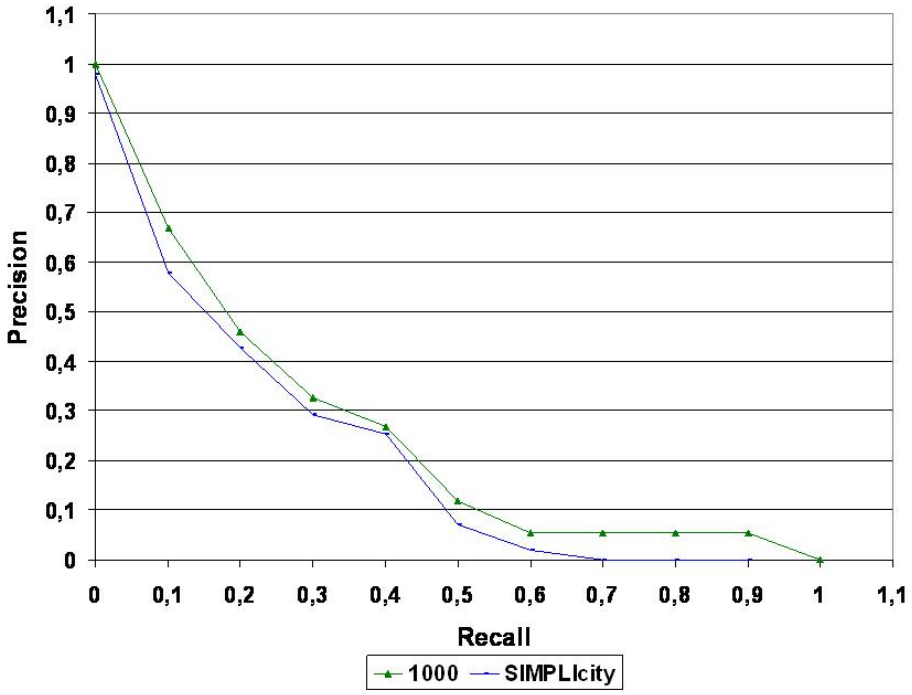


Figure 3. Comparing our technique with Simplicity. In our method we used a lexicon of size 1000 and each region was associated with 10 visual terms. In Simplicity we used the MPEG-7 Scalable Color descriptor, which was the best among the MPEG-7 descriptors. Performance of our approach is better than SIMPLIcity applied to the best MPEG-7 descriptor.

were segmented in at most 10 regions, and we used a lexicon of 1000 visual terms obtained by clustering the regions extracted from a subset of 1000 images. Indexing was performed by giving 10 senses to every visual term.

For the second experiment we used the same subset of COREL used in the Simplicity paper [6]. It consists of 1000 images classified into 10 classes. Each class contains 100 images. An example of images from each class is shown in Figure 2. Every image of the dataset was used as a query and retrieval rank of all remaining images was recorded. A retrieved image was considered a correct match if and only if it was in the same class of the query. The two systems were compared by computing the precision within the first 100 retrieved images. The total number of semantically related images for each query was fixed to 100, that is the size of each class. For this test we have used two different settings for segmentation. We used a fine grained segmentation that returned about 30 regions per images, and a coarse grained segmentation that returned about 10 regions per image. We have also generated two different lexicons containing respectively 100 and 1000 visual terms. We used 5 senses per region with the small lexicon and 10 senses per region with the large lexicon. The Simplicity system was tested with all 5 MPEG-7 descriptor separately.

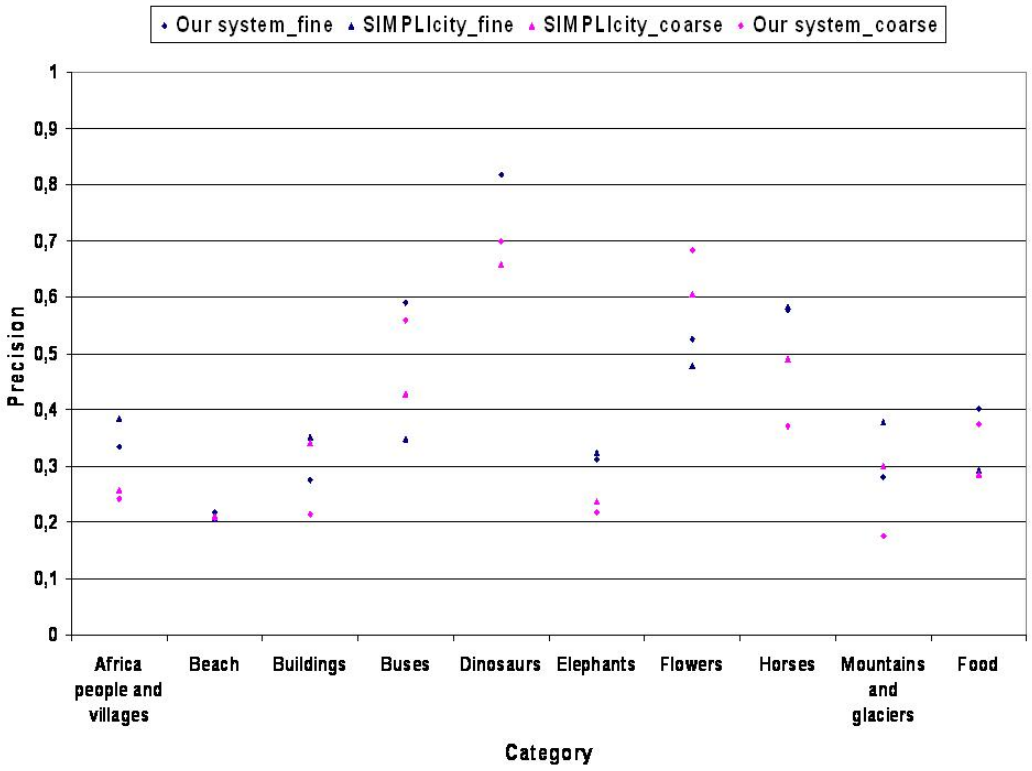
5.2.3 Results

Simplicity obtained the best performance by using the Scalable Color MPEG-7 visual descriptor.

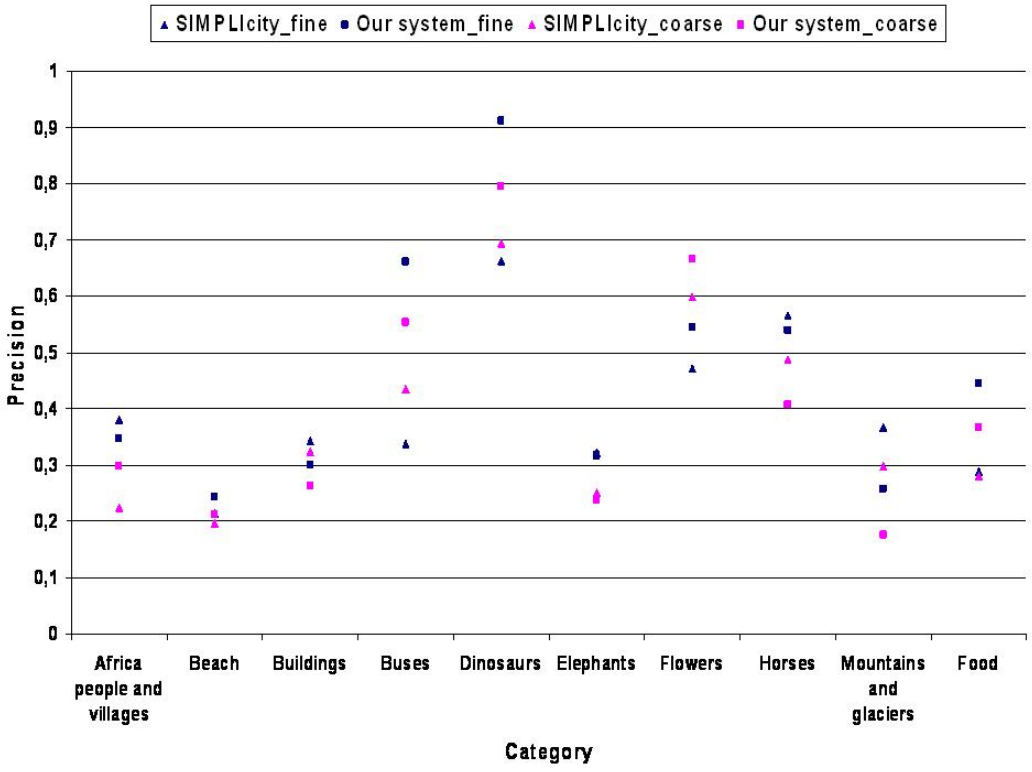
The result of the first test are reported in Figure 3. Here we report the comparison of our approach using the combined (multi-feature lexicon) method and Simplicity with the use of the Scalable Color. As it can be seen, our approach always outperforms Simplicity.

The results of the second test are shown in Figure 4. Here we compare our system and Simplicity, with the use of the Scalable Color descriptor, under various configurations.

It is evident that the result varies in correspondence of different classes. The use of a finer segmentation improves the performances in the both systems but especially in our system, with the exception of Flowers categories for which the results with coarse segmentation are better for both sys-

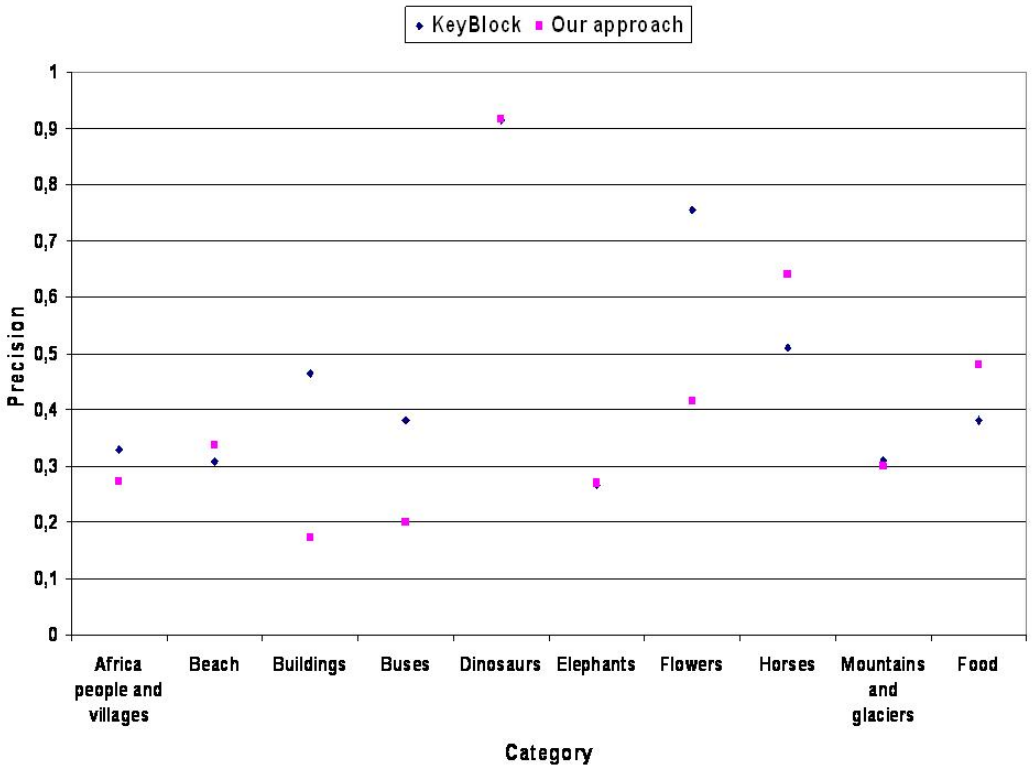


(a) Lexicon of 100 terms

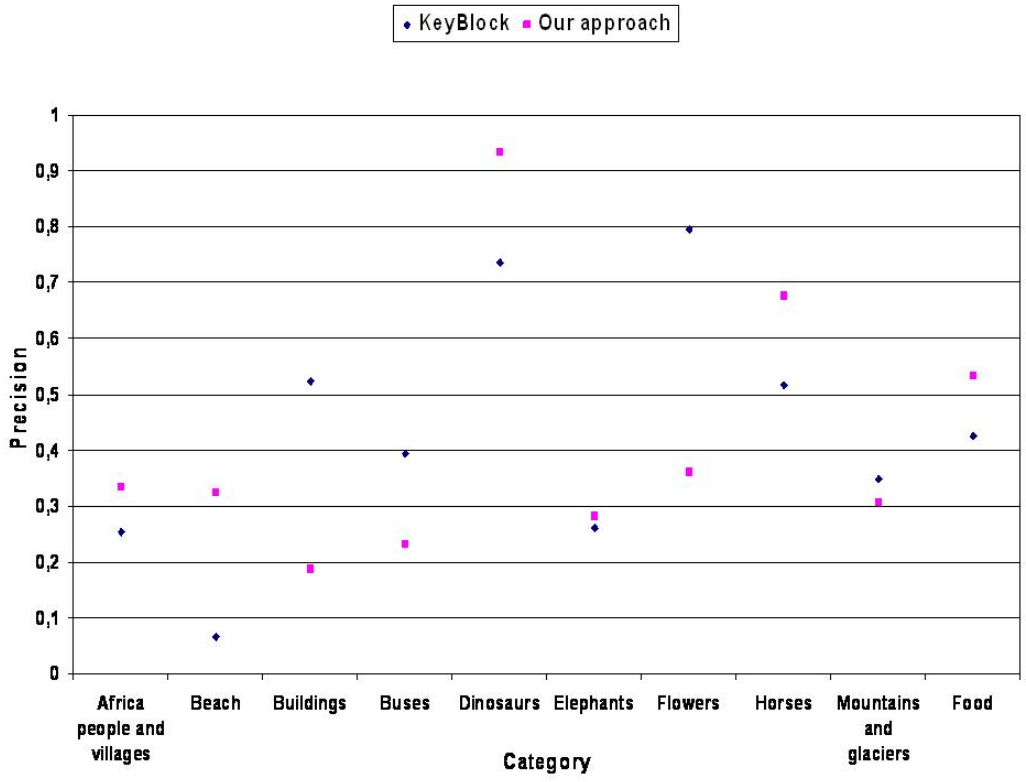


(b) Lexicon of 1000 terms

Figure 4. Comparing our technique of weighting with SIMPLIcity



(a) Lexicon of 256 terms



(b) Lexicon of 1000 terms

Figure 5. Comparing our technique of weighting with Keyblock

tems. Globally we can see that there is not a clear winner between the two systems. In some cases our approach performs better, in some cases Simplicity performs better, in others they almost overlap. However a clear advantage of our approach is that it requires much less storage memory. In fact, Simplicity in order to obtain the score of an image with respect to a query has to compute the similarity among their regions. This means that the descriptors of all regions in the database have to be maintained. In our case, given that images are described in terms of visual terms, and given that just a few visual terms are used (100 or 1000) in our tests, just the visual descriptors of such visual terms should be stored.

5.3 Comparison with Keyblock

Here we compare our approach with that used by the KeyBlock system [2]. In KeyBlock images are segmented by dividing them in blocks of the same size. RGB values of pixels that compose a block are used as features of the block itself.

5.3.1 Description of the experiments

In order to compare our system with keyblock we have again used the 1000 images from the COREL collection as described in previous section and we have performed the test that checks the ability to retrieve images belonging to the same collection, as discussed in previous section.

5.3.2 Experiment settings

Given that the KeyBlock approach strongly depends on the segmentation strategy and feature extraction, we have adopted their proposals also in our system. Thus the images were partitioned into smaller blocks of fixed size. The features extracted for each block corresponds to the RGB value of pixels that compose the block. A subset of the obtained blocks was used to generate the lexicons used in our approach. The tests for the comparison with KeyBlock was performed with block sizes 4X4, given that the authors of the KeyBlock strategy have proved that the performance is good with this block size. Two lexicons were built one of size 256 and the other of size 1000. We have chosen 5 as number of senses for regions for the small lexicon and 10 for the bigger lexicon.

5.3.3 Results

Figure 3 presents the results obtained comparing our approach and KeyBlock, figure 5(a) for the 256 size lexicon and figure 5(b) for the 1000 size lexicon. Figure 5(a) shows that there is not a clear winner in some cases we are better, in others KeyBlock is better, and in others we are almost the same. Increasing the size of lexicon, as shown in Figure 5(b), our system has a clear improvement resulting better in 5 classes, with respect to KeyBlock.

6 Conclusions

We have discussed a proposal for performing image similarity retrieval by means of techniques inspired to text retrieval. The approach is promising given the reduced space required for maintaining necessary data structures and for the possibility of using efficient techniques that have been tested in text retrieval systems. We have performed some preliminary tests of effectiveness and the results seems to be comparable and sometime better than that obtained by other techniques, where much more storage space is required and less efficiency can be obtained.

References

[1] J. Fauqueur and N. Boujemaa. Mental image search by boolean composition of region categories. *Multimedia Tools and Applications*, 31(1):95–117, 2004.

[2] A. Z. Lei Zhu. Theory of keyblock-based image retrieval. *ACM Trans. Inf. Syst.*, 20(2):224–257, 2002.

[3] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis. Still image segmentation tools for object-based multimedia applications. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(4):701–725, June 2004.

[4] P. Salembier, T. Sikora, and B. Manjunath. *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[5] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1983.

[6] J. Z. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture LIbraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.