Indexing Video by the Content

Mohammed Amin Belarbi, Saïd Mahmoudi and Ghalem Belalem

Abstract Indexing video by content represents an important research area that one can find in the field of intelligent search of videos. Visual characteristics such as color are one of the most relevant components used to achieve this task. We are proposing in this paper the basics of indexing by content, the various symbolic features and our approach. Our project is composed of a system based on two phases: an indexing process, which can take long time, and a search engine, which is done in real time because features are already extracted at the indexing phase.

Keywords Indexing · Indexing image by content · Indexing videos by content · Image processing · Video segmentation · Similarity measure

1 Introduction

Nowadays, with the rapid growth of cloud computing use, a lot of applications such as video on demand, social networking, video surveillance and secure cloud storage are widely used [1]. Multimedia database can stores a huge amount images, videos and sounds. These multimedia items can even been used in a professional field (computer aided diagnosis, tourism, education, museum, etc.) or just been used for

M.A. Belarbi (✉)
Faculty of Exact Sciences and Computer Science, Abdelhamid Ibn Badiss University, Mostaganem, Algeria
e-mail: belarbi_mohammed_amin@yahoo.fr

S. Mahmoudi
Faculty of Engineering, University of Mons, 20 Place du Parc, Mons, Belgium
e-mail: said.mahmoudi@umons.ac.be

G. Belalem
Department of Computer Science, Faculty of Exact and Applied Science, Ahmed Ben Bella University, Oran1, Algeria
e-mail: ghalem1dz@gmail.com

S.C. Satapathy et al. (eds.), *Information Systems Design and Intelligent Applications*, Advances in Intelligent Systems and Computing 434,
DOI 10.1007/978-81-322-2752-6_3

personal use (memories, travel, family, events, movie collection, etc.). However, with the exponential development of existing multimedia database, we can point to the fact that standard applications designed to exploit these databases do not provide any good satisfactory. Indeed, the applications generally used to achieve both segmentation and video sequence retrieval are in the most of the cases enable to use and extract efficient characteristics. Multimedia indexing is an essential process used to allow fast and easy search in existing databases. The indexing process allows to minimize the response time and to improve the performances of any information retrieval system. In last years, content-based images retrieval methods ware generally combining images processing methods and database management systems. The latest developments in databases are aiming to accelerate research in multidimensional index, and to improve the automatic recognition capabilities. However, modern techniques in this field have to fundamentally change the way they achieve recognition. Indeed, it is now necessary to repeat the query several times and then synthesize the results before returning an answer, instead of just a single query as before. The rest of this paper is organized as follows: In Sect. 2, we present related researches about the previous standards of image retrieval systems. Section 3 explains the indexing video concepts. Section 4 reports the experimental results. Conclusions are drawn in Sect. 5.

2 Indexing and Research Images by the Visual Content

Image and videos storage and archiving for both TV channels, newspapers, museums, and also for Internet search engines has for a long time been done at the cost of manual annotation step using keywords. This indexing method represents a long and repetitive task for humans. Moreover, this task is very subjective to culture, knowledge and feelings of each person. That is why; there is a real need for research and indexing methods that are directly based on the content of the image [2]. The first prototype system that have attacked the attention of the research community was proposed in 1970. The first image indexing systems by the contents were created in the mid-90s and were generally designed for specialized and mostly closed databases. CBIR (Content-Based Image Retrieval Systems) systems are based on a search process allowing to retrieve an images from an image database by using visual characteristics. These characteristics, which are also called low-level characteristics are color, texture, shape [2].

2.1 Related Works

A CBIR system is a computer system allowing fast navigation and image retrieval in large database of digital images. Much works have already been done in this area. In many fields like commerce, government, universities, and hospitals, where large digital image collections are daily created. Many of these collections are the product of digitization of existing collections of analogue photography, diagrams, drawings, paintings, and prints. Usually, the only way to search these collections was by indexing keyword, or just by browsing. However, the databases of digital images open the way for research based on content [3]. In the following we will present a list of CBIR based systems.

Hirata and Kato [4] proposed an image retrieval system that can facilitates access to the images by query example. In their system, board extraction is carried on user requests. These edges are compared against the database images in a relatively complex process in which edges are displaced or deformed with respect to the corresponding other [5]. They do not provide an image indexing mechanism based on the contents [5].

The QBIC system [6], is one of the most noticeable systems for interrogation by image content, and was developed by IBM. It lets you compose a query based on various different visual properties such as shape, texture and color, that are semi-automatically extracted from images. QBIC use the R^*-tree like as image indexing method [5, 7].

In [8], authors propose an image retrieval system mainly based on visual characteristics and spatial information. For this purpose, they used dominant wavelet coefficients as feature description. Actually, they focused on effective features extraction from wavelet transform rather than the index structure to achieve fast recovery [5].

On the other side, VisualSEEk [9] is a content-based image retrieval system that allows querying by colored regions and spatial arrangement. The authors of this system developed an image similarity function based on color characteristic and space components.

VIPER is an image retrieval system that uses both color and spatial information of images to facilitate the recovery process. They first extract a set of dominant colors in the image and then derive the spatial information of the regions defined by these dominant colors. Thus, in their system, two images are similar in terms of color and spatial information if they have some major groups of the same color falling within the same image space [5].

2.2 Architecture of an Indexing and Search System Images

An image search system by content includes two phases: offline phase for indexing images databases and online phase for research as shown in Fig. 1.

These systems are running in two steps: (i) Indexing: where characteristics are automatically extracted from the image and stored in a digital vector called visual descriptor [10] looks good. (ii) Research: in this phase the system takes one or more requests from user and returns the result, which is a list of images ordered, based on their visual similarity between the descriptor and the query image using a distance measure [10].

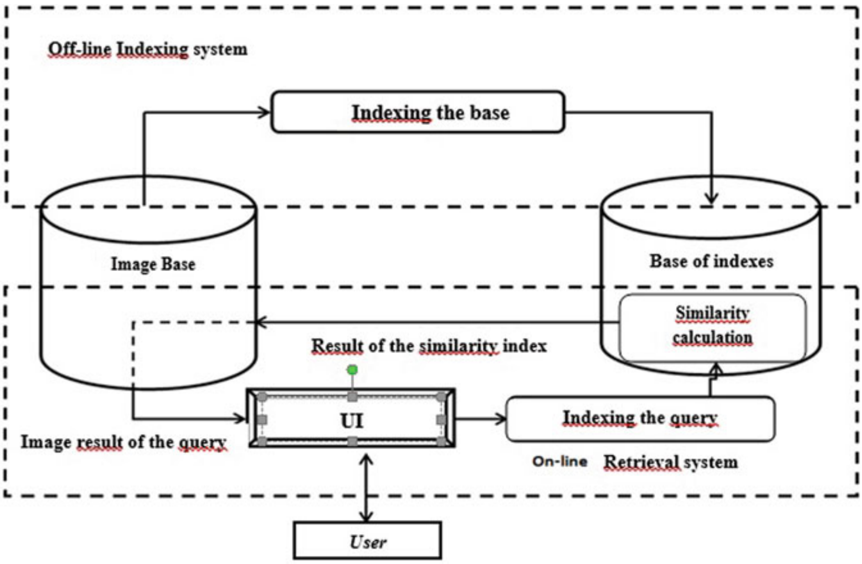

Fig. 1 The architecture of a system of indexing and image search

2.3 Symbolic Features

Colors. Color is the first characteristic that is used for image search. It is the most accessible characteristic because it is directly encoded in each pixel.

The conventional technique for calculating the areas of color is that of the histogram. The idea is to determine the color present in the image and the proportion of the area of the image it fills. The resulting output is called color histogram.

The color histograms can be built in many color ranges, RGB (Red, Green, Blue), the color histogram is produced by cutting the colors of the image into a number of boxes after that counting the number of pixels. Obviously, such a description of the process is simplistic but it is sufficient to search for images of sunset in a collection of maritime or cities images [2, 11].

This technique is often used; Histograms are quick and easy to calculate [12], robust to image manipulation such as rotation and translation [12, 13].

The use of histograms for indexing and image search leads to some problems. (i) Indeed, they have a large sizes, so it is difficult to create a fast and effective indexing using as in [14]. (ii) On the other side, they have no spatial information on colors positions [14]. (iii) They are sensitive for small changes in brightness, which is a problem to compare similar images, but acquired under different conditions [14].

Texture. There is no appropriate definition of texture. However, common sense definition is as follows: the Texture is the repetition of basic elements built from pixels that meet a certain order. Sand, water, the grass, the skin are all examples of textures [2]. It is a field of the image, which appears as a coherent and homogeneous [2].

Various methods are available that are used to describe texture characteristic such as gray-level co-occurrence matrix, Ranklet texture feature, Haar discrete wavelet transform, Gabor filter texture feature, etc. [15].

Shape. The shape is another important visual characteristic, which is however considered a difficult task to be fully automated; the form of an object refers to its profile and physical structure [2].

Shape characteristics are fundamental to systems such as databases of medical image, where the texture and color of objects are similar. In many cases, especially when the accurate detection is necessary, human intervention is necessary [2]. Various methods to extract shape feature are edge histogram descriptor (EHD), Sobel descriptor, SIFT, Fourier descriptor of PFT, etc. [15]. However, in storage applications and retrieving image, shape characteristics can be classified into local and global functions [2].

3 Video Indexing

The objective of a video indexing system is to allow a user to perform a research in a set of videos. Research is conducted on easy criteria to achieve such as the type of emission, actors or theme. But it is extremely complicated and long to get an accurate description of the contents of the videos. Video indexing by content is shown schematically as Fig. 2.

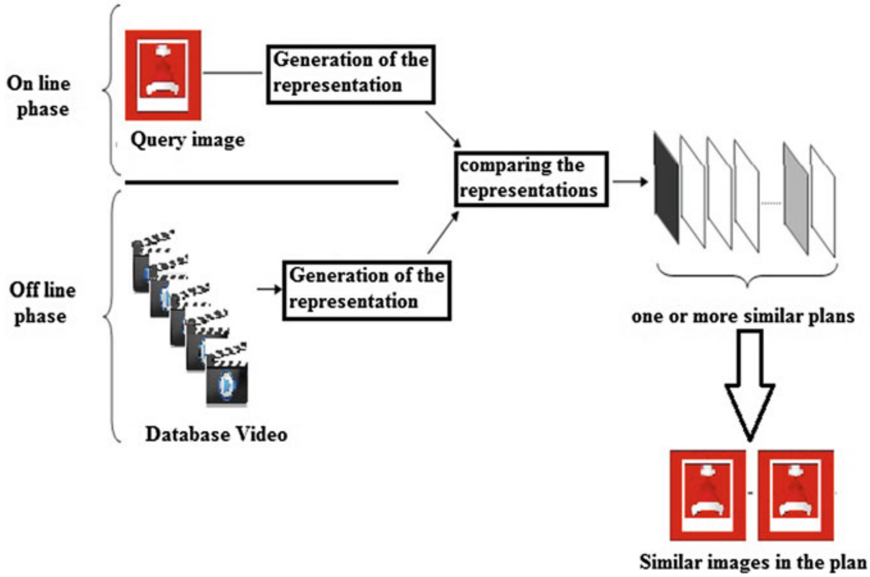

Fig. 2 Architecture of an indexing and retrieval system for video content

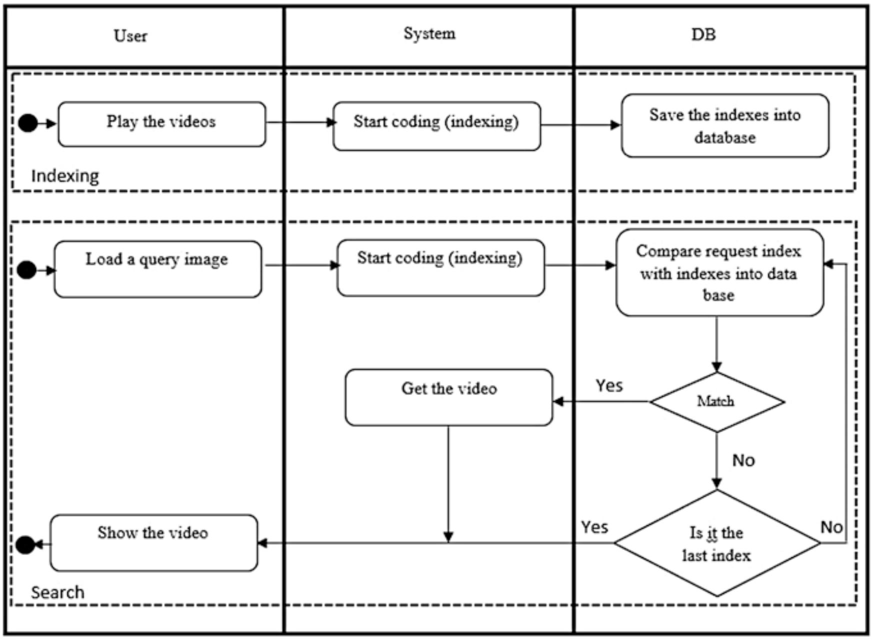

Fig. 3 Activity diagram of our approach

The objective of the proposed video indexing system is to allow a user to search through a set of videos. Figure 3 show the activity diagram of our approach.

- **Images cut**. In this step, we cut a video into a set of individual images. These images allow us to define the following image planes as shown in Fig. 4. After that the following step are processed.
- Application of a visual descriptor images (defined above).
- Calculating a measure of a similarity corresponding to the descriptor applied in the previous step.
- Segmenting a video into several basic units called "plans"; a plan is the shortest unit after the image that defines two shots as shown in Fig. 5.
- Select the key frame. We extract in this step the visual characteristics of each plan, these features are defined in one or more images called "key frames". The key frame of our approach is the first image of each plan because it contains images similar to this image as shown in Fig. 6.

4 Implementation

The general architecture of our application named Videoindexer is shown schematically in Fig. 7, which we developed in C++.

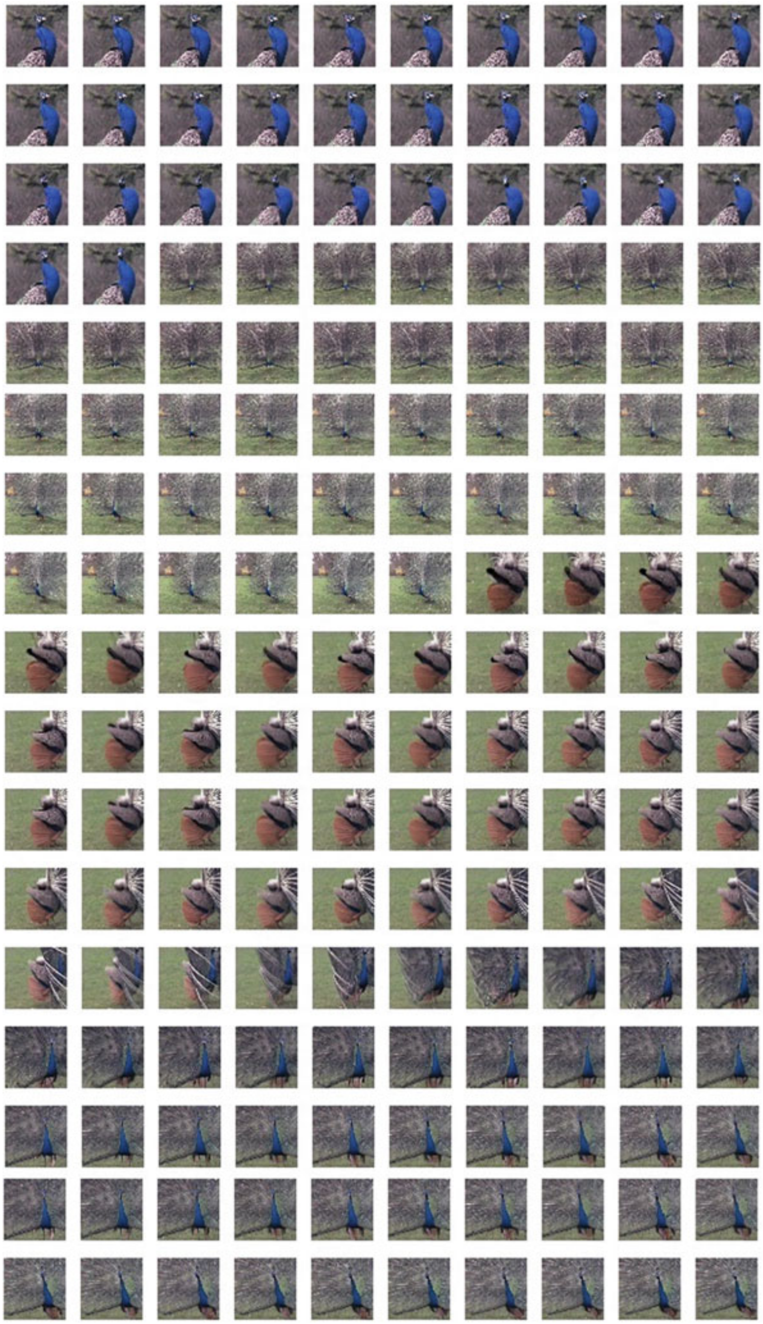

Fig. 4 Cutting into a sequence of images

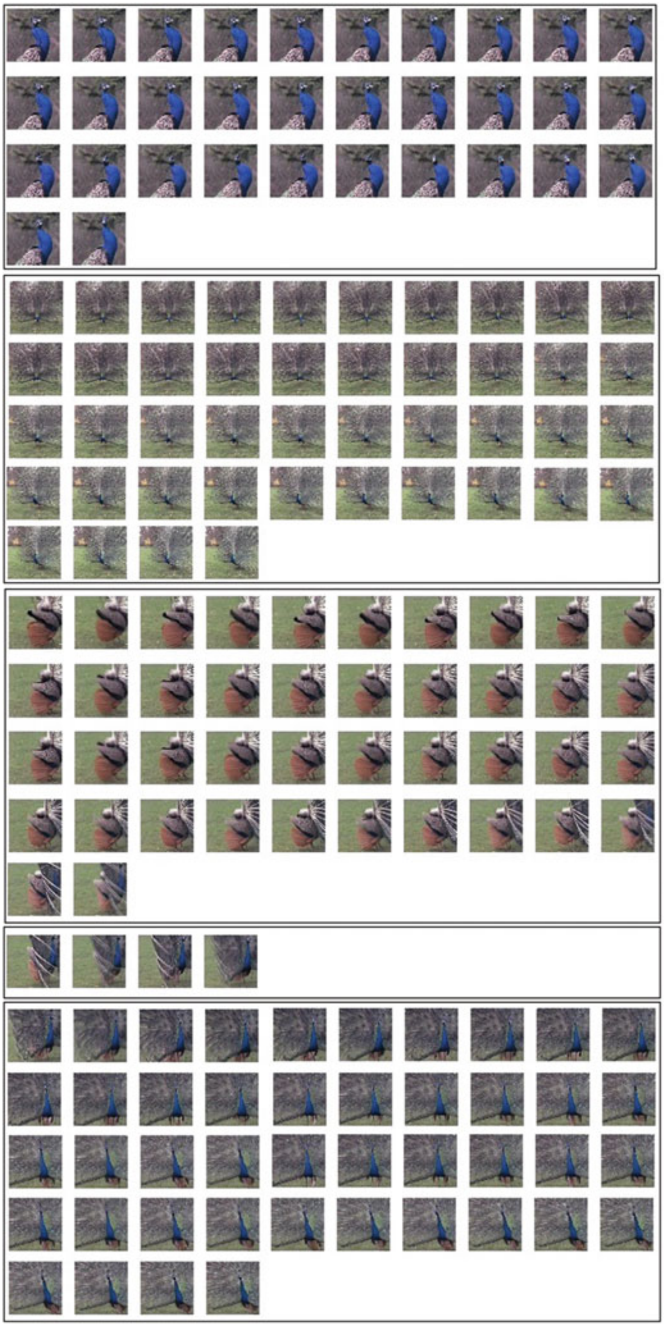

Fig. 5 Temporal segmentation (plan)

Fig. 6 Highlights (key frame of each plan)

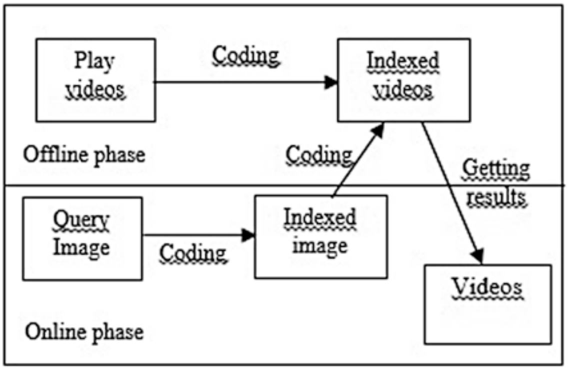

Fig. 7 A general architecture of Videoindexer

4.1 Indexing by the Content

Indexing by the Color. Color is one of the most component used in image retrieval and indexing [16, 17]. It is independent from the size of image and its orientation. In our system, we use the color space: RGB. We chose this space because it is commonly used.

Histogram and distance. Histograms are resistant to a number of changes on the image. They are invariant to rotations, translations [13], and scale changes [14]. But despite that, we must say that the histograms are sensitive to changes in illumination and lighting conditions [11, 14]. We used the histogram correlation technique, Batcharrya [18] and chi-square. With these methods, a distance of similarity is calculated to measure if two histograms are “close” to each other. H_1, H_2 are two histograms.

- Chi-square:

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_{12}(I)} \quad (1)$$

- Correlation:

$$d(H_1, H_2) = \frac{\sum_{i=1}^{N} \bar{H}_1(i)\bar{H}_2(i)}{\sqrt{\sum_{i=1}^{N} \bar{H}_1(i)^2 \sum_{i=1}^{N} \bar{H}_2(i)^2}} \tag{2}$$

$$\bar{H}(i) = H(i) - \frac{1}{N}\sum_{i=1}^{N} H(i) \tag{3}$$

- Bhattacharya:

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1\bar{H}_2N^2}}\sum_{i=1}^{N} \sqrt{H_1(i)H_2(i)}} \tag{4}$$

4.2 Algorithm

Offline phase (Indexing). This phase is summarized in three steps: segmenting videos into individual images, then calculating their histograms and saving these extracted data in a database.

Algorithm 1: (*Indexing the videos*)

Begin:
1. Read Nbvideos;// number of videos
2. **For** i=1 **to** Nbvideos **do {**
3. Read video (i);
4. Read Nbimage; // frame number of the video
5. **for** j=1 **to** Nbimage **do {**
6. Read image (j);
7. Calculate their histogram (Hr, Hg, Hb);
8. Normalize the histogram;
9. Indexing the histogram as XML format and store it in a file;
10. Store the image (j) in a jpeg file;
11. Storing in a DB the image path (j) and the path of XML file; **} }**
End

In line 4 of Algorithm 1, we read the frame number of the video, the number of images depends on the length of the video. Also, we calculate the histogram for each image then we must normalized it as in line 7. Then we store the histogram as XML format as in line 9.

Online phase. This is the second part of our approach

Algorithm 2: (Research the videos)

```
Begin:
1. Read the query image;
2. Read the threshold;
3. Choose a distance;
4. Calculate the histogram of the query image (H1R, H1G, H1B);
5. Normalized the histogram of the query image (H1R, H1G, H1B);
6. Read the content of the database;
7. While there are records in the database do {
8.        for i=1 to the last record into the database do {
9.             Read ith histogram recording (H2R, H2G, H2B)
10.            Retrieve all videos, which distance (h1, h2) < threshold } }
End
```

In line 3 of Algorithm 2, we choose a distance between the distances defined above such as chi-square. Also in line 10, we retrieve all the videos to display them to the user.

The research phase is quick, which due to the sequential search, and also because in this phase, we do not process any treatment of video and the image. We calculate just the histogram of the query image, and after that the system returns all the similar videos containing the image query.

We notice that our approach allows to save research time because we compare the query image with the key frames as shown in Fig. 8.

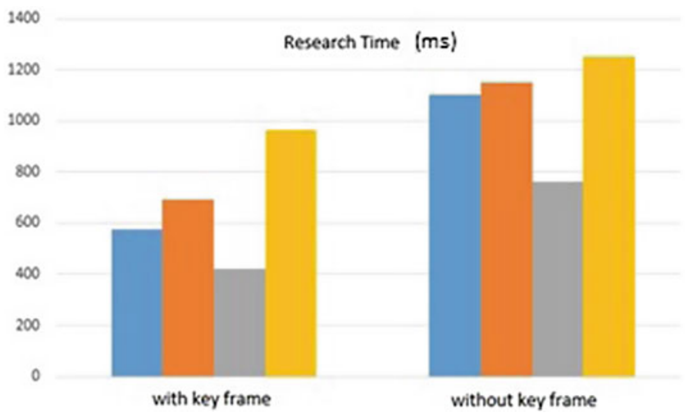

Fig. 8 The research time with key frame and without it

5 Conclusion

In this Paper, we have shown and comment experimental results obtained by the different strategies for calculating similarity. We found that these results positively guarantee the effectiveness of our approach and that a good video indexing process is the one that allows you to find the most relevant correspondence with the least possible number of calculations, and less time. The results obtained from the application allows us to judge and say that the histogram is effective for the overall comparison of the content of the image, simple, response time is fast.

In the future, we propose to improve our system to apply this attribute to image regions (to apply a space research) and combine other descriptors such as SURF, SIFT to improve the relevance of our system.

In our next works, we plan to discuss indexing video when data increase (Big Data), and we will try to reduce the dimensionality to accelerate research phase (response time).

References

1. Cheng, B., Zhuo, L., Zhang, J.: Comparative Study on Dimensionality Reduction in Large-Scale Image Retrieval. In: 2013 IEEE International Symposium on Multimedia. pp. 445–450. IEEE (2013).
2. Idris, F., Panchanathan, S.: Review of Image and Video Indexing Techniques. J. Vis. Commun. Image Represent. 8, 146–166 (1997).
3. Dubey, R.S., Student, M.T.C.S.E., Bhattacharjee, J.: Multi Feature Content Based Image Retrieval. 02, 2145–2149 (2010).
4. Hirata, K., Kato, T.: Query by visual example. In: Advances in Database Technology EDBT'92. pp. 56–71 (1992).
5. Lee, D.-H., Kim, H.-J.: A fast content-based indexing and retrieval technique by the shape information in large image database. J. Syst. Softw. 56, 165–182 (2001).
6. Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., Equitz, W.: Efficient and effective querying by image content. J. Intell. Inf. Syst. 3, 231–262 (1994).
7. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafher, J., Lee, D., Petkovie, D., Steele, D., Yanker, P.: The QBIC System. (1995).
8. Jacobs, C.E., Finkelstein, A., Salesin, D.H.: Fast multiresolution image querying. In: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. pp. 277–286 (1995).
9. Smith, J.R., Chang, S.-F.: VisualSEEk: a fully automated content-based image query system. In: Proceedings of the fourth ACM international conference on Multimedia. pp. 87–98 (1997).
10. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. ACM Trans. Multimed. Comput. Commun. Appl. 2, 1–19 (2006).
11. Rasli, R.M., Muda, T.Z.T., Yusof, Y., Bakar, J.A.: Comparative Analysis of Content Based Image Retrieval Techniques Using Color Histogram: A Case Study of GLCM and K-Means Clustering. Intell. Syst. Model. Simulation, Int. Conf. 0, 283–286 (2012).
12. Zhang, Z., Li, W., Li, B.: An improving technique of color histogram in segmentation-based image retrieval. 5th Int. Conf. Inf. Assur. Secur. IAS 2009. 2, 381–384 (2009).

13. Krishnan, N., Sheerin Banu, M., Callins Christiyana, C.: Content based image retrieval using dominant color identification based on foreground objects. Proc. - Int. Conf. Comput. Intell. Multimed. Appl. ICCIMA 2007. 3, 190–194 (2008).
14. Marinov, O., Deen, M.J., Iniguez, B.: Charge transport in organic and polymer thin-film transistors : recent issues. Comput. Eng. 152, 189–209 (2005).
15. Choudhary, R., Raina, N., Chaudhary, N., Chauhan, R., Goudar, R.H.: An Integrated Approach to Content Based Image Retrieval. 2404–2410 (2014).
16. Takumi Kobayashi and Nobuyuki Otsu: Color Image Feature Extraction Using Color Index Local Auto-CorrElations. Icassp. 1057–1060 (2009).
17. Xue, B.X.B., Wanjun, L.W.L.: Research of Image Retrieval Based on Color. 2009 Int. Forum Comput. Sci. Appl. 1, 283–286 (2009).
18. Aherne, F.J., Thacker, N. a., Rockett, P.I.: The Bhattacharyya metric as an absolute similarity measure for frequency coded data. Kybernetika. 34, 363–368 (1998).