International Journal of Sciences and Techniques of Automatic control & computer engineering
IJ-STA, Volume 3, N° 1, July 2009, pp. 956–971.

A study of the color-structure descriptor for shot boundary detection

Abdessalem Ben Abdelali [1], Mohamed NidhalKrifa [1], Lamjed Touil [1], Abdellatif Mtibaa [1], Elbey Bourennane [2]

[1] Laboratory EμE, Faculty of Sciences of Monastir, Monastir, Tunisia
Abdessalem.BenAbdelali@enim.rnu.tn, kmnidhal@yahoo.fr, lamjedtl@yahoo.fr Abdellatif.mtibaa@enim.rnu.tn

[2] Laboratory LE2I, University of Burgundy, Dijon, France
ebourenn@u-bourgogne.fr

Abstract. *This paper focus on the study of the color structure descriptor (CSD) for shot boundary detection in video sequences. We interest in the validation and the optimisation of this descriptor in the aim of its real time implementation on hardware architecture. In this context, the CSD was applied for 256 and 32 quantification levels in the HMMD color space. The use of a lower number of quantification levels can assure a better computing performance while preserving a satisfactory level of accuracy in term of shot boundary detection rate. This can be useful for constrained applications and platforms. An example of hardware implementation architecture of the CSD is also presented in this paper.*

Keywords. *Color Structure Descriptor (CSD), MPEG7, shot boundary detection, real time implementation.*

1. Introduction

Indexing and retrieval of video becomes more and more important as the size of video data increases. In fact, recent advances in multimedia compression technology, coupled with the significant increase in computer performance and the growth of internet, have led to the widespread use and availability of digital video. This makes the retrieval of desired information more difficult, as more and more data has to be searched. In this context video understanding and semantic information extraction represent important steps towards more efficient manipulation and retrieval of visual media. Based on these concepts, video indexing has as objective to create tools that facilitate research and navigation in multimedia documents. It is used for various purposes such as image understanding (surveillance, intelligent vision, smart cameras, etc.), information retrieval (quickly and efficiently searching for various documents

This paper was recommended for publication in revised form by the editor Staff.
Edition: CPU of Tunis, Tunisia, ISSN: 1737-7749

of interest to the user) and filtering (to receive only those multimedia data items which satisfy the user's preferences).

The increased availability and usage of digital video lead to a need for automated video content analysis techniques. Several methods aiming at automating this time and resource consuming process have appeared in literature [1–3]. A standard, called MPEG-7, has also been elaborated [4]. The standard specifies a set of visual descriptors that include color, texture, shape and motion [5, 6]. Most standardized descriptors are low-level arithmetic ones, chosen so as to ensure their usefulness in wide range of possible applications [7]. Low-level features serve as a basis for a number of analysis algorithms for higher level features (such as structuring or classification) or they can be used directly for similarity matching based on these descriptors [8].

Low-level visual features extraction defines the process of creating a descriptor from a given visual media item. This process may include complex algorithms that can be also combined in a specific scheme to be used as an input for the extraction of higher level features. Complex algorithms for classification, recognition or summarization are also used. In addition, specialized tools and systems for information analysis, filtering, and managing not only must work with data that has been previously stored, but also with live data being broadcasted through high-speed means such as digital cables. These considerations must be taken in account when designing a system for content based video indexing. The target hardware for such system must satisfy different applications needs. In fact, the multimedia treatments are actually emerged in many different kinds of systems. These systems have demanding applications that can be driven by portability, performance, processing power, cost, etc [9-11]. To satisfy the requirements of such constrained applications, hardware acceleration techniques can be used.

In this study we interest in the validation and the optimisation of the Color Structure Descriptor (CSD) in the aim of its possible real time implementation on hardware architecture. We have applied this descriptor for 256 and for 32 quantification levels in the HMMD color space. We have considered its application for still image retrieval and for shot boundary detection which represents the first step towards automatic annotation of digital video sequences. The use of a lower number of quantification levels is applied in order to assure a complexity reduction and to speed up the algorithm.

The rest of this paper is organized as follows: in section 2, we present the CSD specification. In section 3 and section 4 we focuses respectively on the study of the CSD for still image retrieval and for shot boundary detection in video sequences. The different experimentations and the evolution results of various application modes of the CSD are presented in these two sections. An example of hardware architecture for this descriptor is presented in section 5.

2. The Color Structure Descriptor (CSD)

The color structure descriptor [5, 12-14] represents an image or an image region by its color distribution. The main function of this descriptor is image to image matching and its intended use is for still-image retrieval. It provides satisfactory image indexing and retrieval results among all color-based descriptors in MPEG-7 standard [15]. The superiority comes from the consideration of space distribution of colors. The CSD expresses the local color structure in an image using an 8x8 structuring element. In fact, instead of characterizing the relative frequency of individual image samples with a particular color, this descriptor characterizes the relative frequency of structuring elements that contain an image sample with a particular color. Hence, unlike the color histogram, this descriptor can distinguish between two images in which a given color is present in identical amounts but where the structure of the groups of pixels having that color is different in the two images. Fig 1 [14] shows two different images with two iso-color planes: grey and black. Using a standard color histogram, the two images are described as identical, as they have the same number of black and grey pixels. However, as we can see the structure of the two images differs significantly.

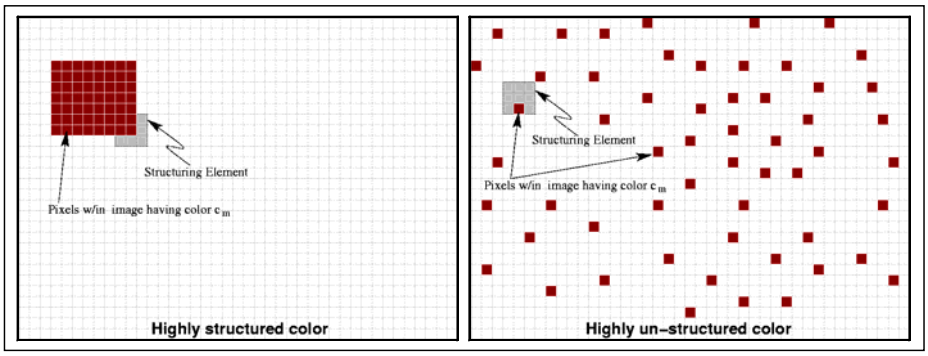

Fig. 1. Two images having the same traditional histogram, but the right one has much more grey components in CSD description

The CSD is identical in form to a color histogram but it is different in term of semantics. Suppose the number of colors represented in a histogram is denoted by M; that is, the colors in the image are quantized into M different colors c_0, c_1, c_2, …, c_{M-1}. The color structure histogram can be denoted by h(m), m=0, 1, …, M-1, where the value in each bin represents the number of structuring elements in the image containing one or more pixels with color "C_m". The final CSD is represented by 1D array of 8-bit quantized values. The performance of this descriptor increases considerably while using the HMMD color space [5].

Fig 2 depicts the different CSD extraction steps. The Color Structure descriptor shall be defined using the HMMD color space. The color pixels of incoming images in any other color space shall be converted to the HMMD color space and requantized appropriately before extracting the structure histogram. The CSD is defined using

four color space quantization operating points: 256, 128, 64, and 32 bins [5]. The number of quantization levels constitutes also the number of the histogram bins. The CSD containing 256 bins is directly extracted from the image based on a 256-cell quantization of the HMMD color space. For a bin number less than 256, the bins can be computed based on a unification of the bins of the 256-bin descriptor.

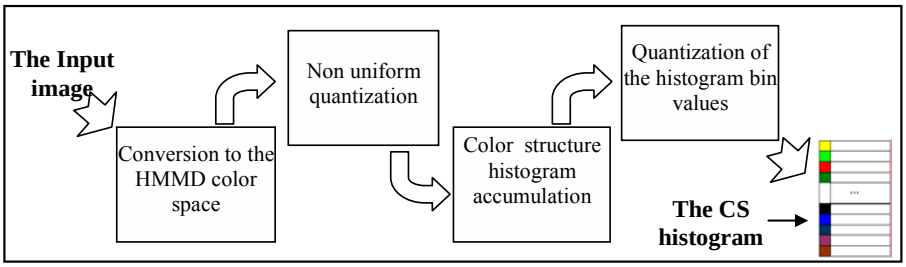

Fig. 2. The extraction steps of the color structure descriptor

The Color Structure histogram accumulation step is illustrated by the example of Fig 3 [14]. The Histogram is computed by visiting all locations in the image, observing which colors are presented in it, and then updating color bins by adding one, no matter how many the same color pixels exist. The increase (or not) of the bins is determined by the presence (or absence) of the corresponding colors, and not by the count of the pixels of each present color. Therefore, in any given position of the structuring element, the increase can be either 0 or 1. For example, suppose we have 8 different colors as in Fig 3 and the structuring element has a size of 8 by 8 pixels. In the location represented in fig 3, the structuring element contains some pixels with color c1, some pixels with color c3 and some pixels with color c6. Then, the bin labeled c1, the bin labeled c3 and the bin labeled c6 would each be incremented once. So, in this location, the Color Structure histogram is incremented three times, once for each color present in the structuring element area.

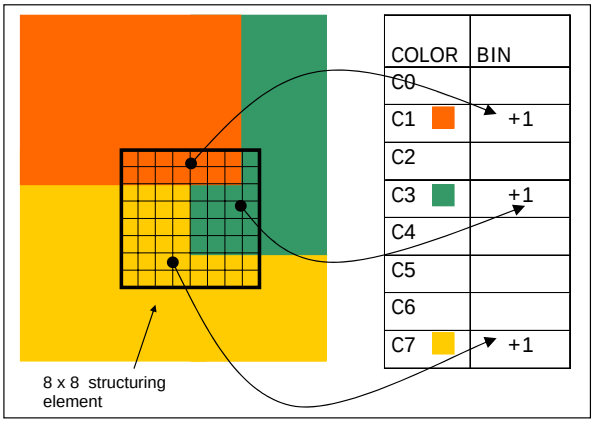

Fig. 3. Color structure histogram accumulation

The spatial extent of the structuring element depends on the image size but the number of samples in the structuring element is held constant by subsampling the image and the structuring element at the same time. The number of samples in the structuring element is always 64, and the distance between two samples in this pattern increases with the image size as shown in Fig 4. The following rule determines the spatial extent of the structuring element (equivalently, the sub sampling factor) given the image size:

$$K = 2^{P}$$
$$P = \max\left\{0, round\ (0.5 * \log_2(width\ * height\ ) - 8)\right\}$$
$$E = 8 * K$$

- E : the spatial extent of the structuring element in the original image.
- K : the subsampling factor (K= 1, 2, 4, 8, 16, …). Where *K*=1 implies no subsampling, *K*=2 implies subsampling by 2 horizontally and vertically, etc. For example, an image of the size 320x240 yields *K*=1 and *E*=8, an image with a sizes of 640x480 yields *K*=2 and *E*=16.

Fig 4 shows the structuring element in the initial location at the upper left corner of the image. The structuring element slides over the image and is shifted by one pixel in case (a) and by two pixels in case (b). Case (b) corresponds to a subsampling of the image by 2 in both directions and subsequently applying the same 8x8 structuring element.

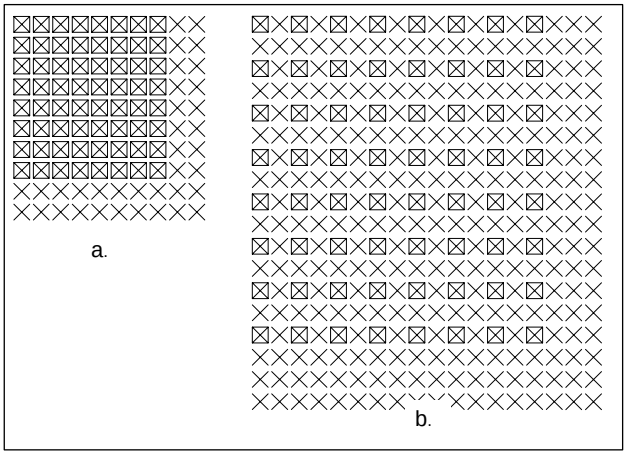

Fig. 4. Structuring elements for images with different resolutions: (a): 320x240, (b): 640x480

The final step of the extraction process is the non-uniform quantization of each bin amplitude to an 8-bit *code value*. Before this quantification, bin amplitudes are normalized by R_{max} . R_{max} represents the number of positions within the (possibly subsampled) image that the structuring element can occupy if its origin is moved to every allowable location. For example, for a 320x240 image the normalizing factor is (320-7)x(240-7). Normalized bin amplitude values lie in the range [0.0, 1.0].

The matching procedure determines the similarity of two visual items by computing the similarity between their color-structure histograms. Let $\mathbf{h}_A(i)$ be the histogram vector of visual item A, and $\mathbf{h}_B(i)$ be the histogram vector of visual item B, the L_1 norm histogram distance is given by:

$$\mathrm{dist}(A,B) = \sum_i \left|\mathbf{h}_A(i) - \mathbf{h}_B(i)\right|$$

The image with the lowest histogram-distance with the query image is considered to be the image that best matches the query image in terms of color.

3. Application of the CSD for content based-image retrieval

One of the common uses of MPEG-7 descriptors is content based image retrieval. Fig 5 represents the scheme of an image retrieval system principle. A sample image is presented to the retrieval system in order to find images, in the data base, with similar visual characteristics. The query image is processed and different MPEG-7 descriptors are extracted. These descriptors will be sent to the retrieval engine. This module (retrieval engine) compares the description given by the user with those already available in the data base and returns to the user interface a sorted list of the images. Such list is then presented to the user in form of thumbnails, to ease the navigation in the database.

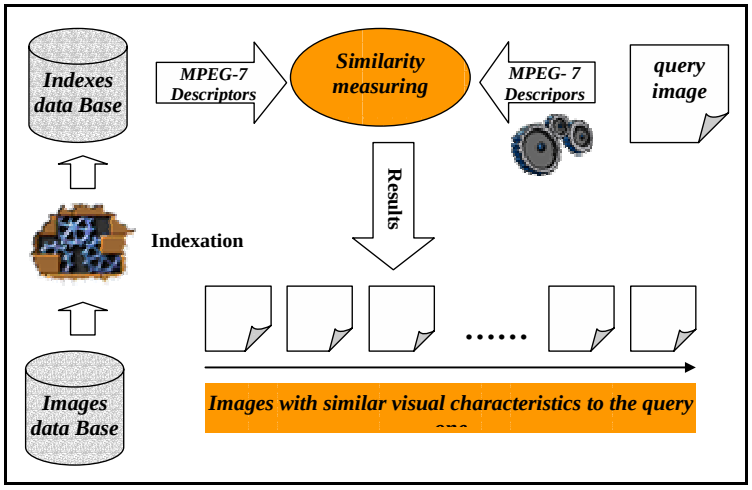

Fig. 5. Principle of content based image retrieval systems

To evaluate the color structure descriptor we have used a rich corpus of different image genres. An example of retrieval results using the CSD with 256 and 32 quantification levels is given respectively in fig 6 and fig 7. The obtained results demonstrate the performance of the CSD for color based image retrieval either for 256 or 32 quantification levels.

Fig. 6. Example of image retrieval results with the CSD descriptor – 256 quantification levels

Fig. 7. Example of image retrieval results with the CSD descriptor – 32 quantification levels

4. Application of the CSD for temporal video segmentation

4.1. Shot boundary detection

Shot boundary detection is a fundamental task in any kind of video content manipulation. Task is to identify the shot boundaries with their location and type in the given video clip(s). In produced video such as television or movies, shots are separated by different types of transitions, or boundaries. Transition effects can be classified into two major categories: cut or abrupt shot change and gradual transition such as fades and dissolves. In the following we are interest only in the CUTs detection that generally represents more than 75% of video shot boundaries.

In the case of a CUT, the last frame of the first video sequence is directly followed by the first frame of the second video sequence. A hard cut can be defined as the direct concatenation of two shots S1(x,y,t) and S2(x,y,t). No transitional frames are involved. Thus the resulting sequence S(x,y,t) is formally given by :

$$S(x,y,t) = (1\text{-}u\text{-}1(t\text{-}tcut))S1(x,y,t) + u\text{-}1(t\text{-}tcut)S2(x,y,t)$$

Where tcut denotes the time stamp of the first frame after the hard cut and u-1(t) the unit step function (1for >=0, 0 else).

The application of the CSD for CUT detection consists in extracting this descriptor for each frame in the considered video sequence and comparing the obtained characteristic vector for consecutive frames. The basis of this method consists in detecting visual discontinuities along the time domain. Suppose g (n, n+K) the measure of similarity degree between two frames based on the visual characteristic that is given by the used descriptor. This measure is related to the difference or discontinuity between frame n and n+k where k ≥ 1. If the obtained distance exceeds a threshold a shot change will be reported. This principle is illustrated by Fig 8.

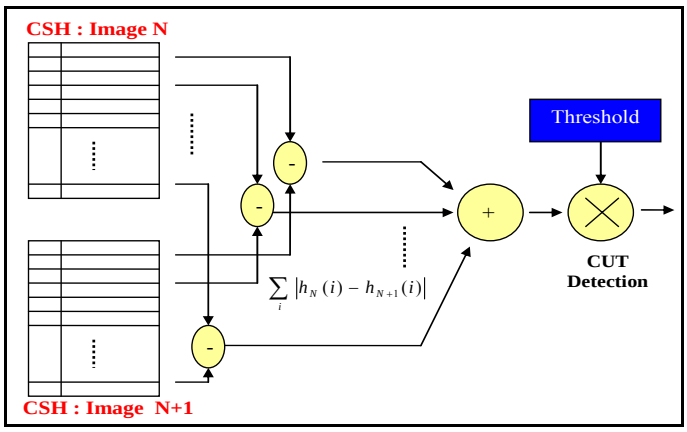

Fig. 8. Utilization of the CSD for CUTs detection

4.2. Video corpus

To evaluate the CSD for shot boundary detection we have exploited a video corpus of 36 video sequences with a total of 56141 frames. The test data comprises media clips from six different genres: advertisements, cartoons, documentaries, movies, football and news. The number of CUTs is 461 (79.35%) and the number of progressive transition is 120 (20.65%). The shot transitions and their positions are manually determined to be used for the verification of the results obtained by the automatic methods. Fig 9 shows example frames from the clips employed in the evaluation.

Fig. 9. Example frames from the clips employed in the experimental evaluation

4.3. Experimentation and evaluation results

To evaluate the performances of the shot boundary detection techniques many evaluation measures can be used [16 -18]. The most adapted are the following: "Precision", "Recall", "False positives (FP)".

$$\Pr ecision = \frac{\text{Transitions Correctly Reported}}{\text{Transitions Reported}}$$

$$\text{Re}\,call = \frac{\text{Transition s Correctly Reported}}{\text{Transition s in Reference}}$$

$$FP = \frac{\text{False Transition s}}{\text{Transition s Reported}}$$

For every video sequence we determine the number of shots, the number of shots correctly reported, the number of false detections and the number of non reported shots. For each sequence we also draw the curve of the distances between the successive frames. These curves are mainly used to determine the threshold values, but they also give an idea about the capacity of the used technique in detecting transitions.

We have evaluated the CSD for 256 and 32 quantification levels in the HMMD color space. The HMMD color is quantized non-uniformly as follows [11]. First, the HMMD color space is divided into 5 subspaces. This subspace division is defined

along the Diff axis of the HMMD color space. The subspaces are defined by cut-points which determine the following diff-axis intervals: [0,6), [6, 20), [20, 60), [60, 110) and [110, 255]. Second, each color subspace is uniformly quantized along the Hue and Sum axes. The number of quantization levels along each axis is presented in Table 1 for 256 and 32 histogram bins.

Table 1. HMMD color space quantization for Color Structure descriptor

Subspace	Number of quantization levels for 256 and 32 histogram bins			
	256		32	
	Hue	Sum	Hue	Sum
0	1	32	1	8
1	4	8	4	4
2	16	4		
3	16	4	4	1
4	16	4	4	1

Erreur ! Source du renvoi introuvable. 10 shows a slice of the HMMD space in the diff-sum plane for zero hue angle and depicts the quantization cells for the 32-cell operating point. Cut-points defining the subspaces are indicated in the figure by vertical lines in the color plane. The diff-axis values that determine the cut-points are shown in black at the top of the dashed cut-point markers along the upper edge of the plane. Horizontal lines within each subspace depict the quantization along the sum-axis. The quantization of hue angle is indicated by the gray rotation arrows around each cut-point marker. The gray number to the right of a rotation angle corresponds to the number of levels to which hue has been quantized in the subspace to the right of the cut-point.

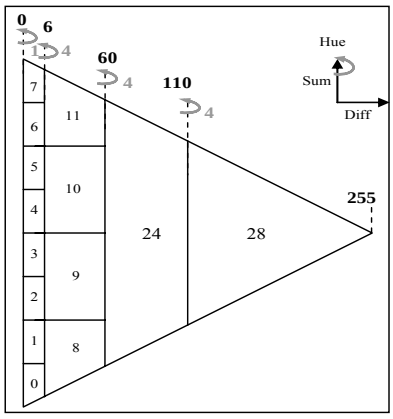

Fig. 10. Correspondence between 32-cell HMMD color space and bin indices

The obtained results for the different evaluation tests are given in Table 2. This table gives the performances (precision, recall and false positives) of the CSD for 256 and 32 quantification levels and for different video sequence types.

Table 2. Performances of the CSD for 256 and 32 HMMD quantification levels

	Precision		***Recall***		***FP***	
	256 levels	*32 levels*	*256 levels*	*32 levels*	*256 levels*	*32 levels*
News	*97 %*	*97 %*	*100 %*	*100 %*	*3 %*	*3 %*
Football	*97 %*	*97 %*	*100 %*	*97 %*	*3 %*	*3 %*
Documentaries	*85 %*	*86 %*	*94 %*	*94 %*	*15 %*	*14 %*
Films	*87 %*	*90 %*	*97 %*	*90 %*	*13 %*	*10 %*
Advertisements	*74 %*	*78 %*	*63 %*	*64 %*	*26 %*	*22 %*
Cartoons	*79 %*	*83 %*	*82 %*	*82 %*	*21 %*	*17 %*

An example of graphical results (for the video “News 4”) is given in Fig 11. The curve represented in this figure demonstrates the good capability of the CSD in shot boundary detection for both quantification levels: 256 and 32.

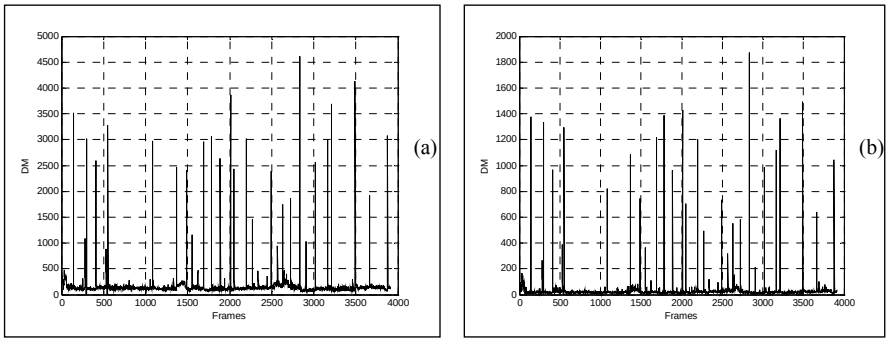

Fig. 11. Graphical result for the video "News 4" (a: 256 HMMD levels, b: 32 HMMD levels)

In parallel to the shot boundary detection the first frame of each shot is considered as the key image of the detected shot. The obtained key frames are used for video abstraction. Fig 12 represents an example of still based abstract corresponding to the sequence "News 2".

5. Example of architecture for hardware implementation of the CSD

Fig. 12. An example of still-image based abstract (the"News 2" video sequence)

In this section we present an example of hardware implementation solution for the CSD. We are interested only in the color structure histogram block which represents the main step in the CSD extraction module (Fig 14).

Firstly we consider that the pixel scan order of the structuring window is done as following: After finishing a stripe we move to the next one until all stripes in a frame are visited. The displacement of the structuring window between adjacent stripes is of one pixel. The histogram of color structures is calculated in two steps: recording the local histogram which indicates the number of appearances of each color in the structuring element and updating the color structure histogram (incrementing each histogram bin which correspond to a color that appears in the structuring element). A one bit register bank is used to indicate which colors exist in the structuring element (non-zero bin means the color belongs to the window) to update the colors structure histogram. This principle is illustrated by Fig 13.

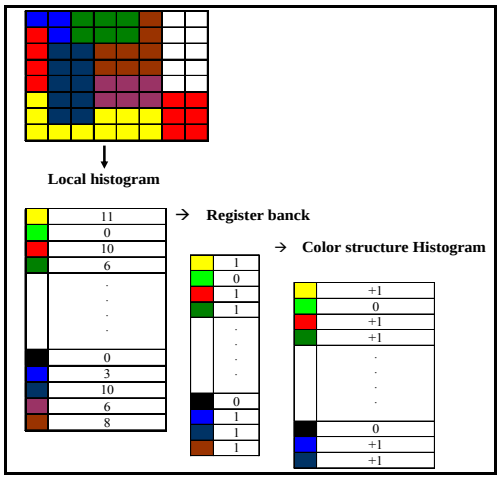

Fig. 13. Principle of the colors structure histogram

The CSD block diagram is shown in Fig 14. The clor structure histogram module is based on the block named local histogram. The architecture of this block is shown by Fig 15. It contains an SRAM to record the local color histogram and a register bank to indicate which colors exist in the structuring element. The SRAM gets the address from the quantized input color to update the corresponding bin in the local histogram. The input color is also used to select the corresponding register. When the local histogram block receives a new HMMD value it updates the bin corresponding to this value and in the some time it turn in to one the bit that indicates the existence of this color value in the structuring element. In passing from one structuring element position to the following one only an updating of the local histogram will be carried on (we do not have to reread all the 64 color values). The local histogram updating consists in adding to the corresponding bins the number of colors belonging to the new structuring element and subtracting those which belong to the previous one and are not a part of the current one. The updating of the colors structure histogram values is simply done by adding the values of the 1 bit registers (0 or 1) to the corresponding bins.

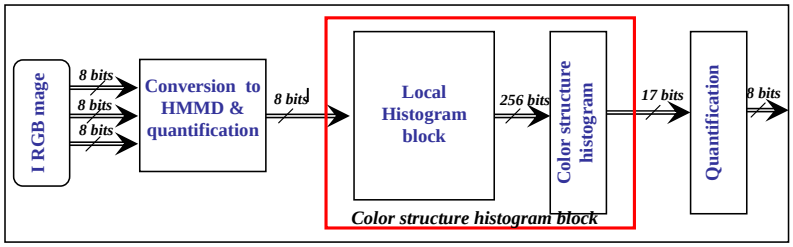

Fig. 14. The CSD block diagram

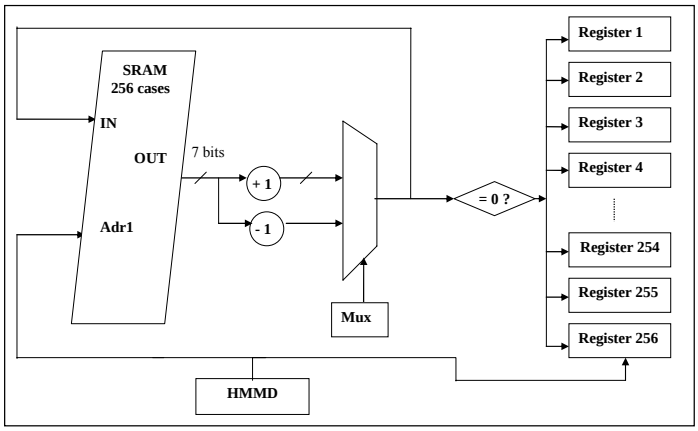

Fig. 15. The local histogram block

7. Conclusion

This paper brought an experimental study of the CSD particularly for shot boundary detection. After introducing the content based video indexing problem, we have presented the specification of the CSD in section 2. Section 3 and section 4 were dedicated to present the experiments of the CSD for still images retrieval and for shot boundary detection in video sequences. The CSD was applied for 256 and for 32 quantification levels. Experimental results demonstrate that the use of low number of quantification levels can assure a satisfactory level of accuracy in term of shot boundary detection rate. This can promise a better computing performance and can be useful for real time implementation on resource constrained platform. In the last section of this paper an example of hardware implementation architecture of the CSD was presented. Different other transformations of the CSD algorithm are experimented. The proposed transformations intend to optimise the CSD in the aim of assuring a better computing performance for real time implementation.

References

1. A. Cavallaro and T. Ebrahimi: Interaction between High-Level and Low-Level Image Analysis for Semantic Video Object Extraction. EURASIP Journal on Applied Signal Processing, Volume (2004), Issue 6, pp. 786-797, 2004
2. Alan Smeaton, Paul Over: Shot Boundary Detection Task Overview. TRECVID-2006

3. Ying Li, Tong Zhang and Daniel Tretter: An Overview of Video Abstraction Techniques. Document HPL-2001-191, Imaging Systems Laboratory, HP Laboratories Palo Alto, July 31st 2001

4. Chang SF, Puri A, Sikora T, Zhang H. Overview of the MPEG-7 standard. IEEE Trans Circ Syst Video Technol 2001;11:688–95

5. B. S. Manjunath, Jens-Rainer Ohm, Vinod V. Vasudevan and Akio Yamada: Color and Texture Descriptors. IEEE Transactions on Circuits and Systems for Video Technologiy, Vol. 11, NO. 6, pp. 703-715, June 2001

6. S. Jeannin and A. Divakaran: MPEG-7 visual motion descriptors. IEEE Tr. CSVT, vol. 11, pp. 720–724, June 2001.

7. Mezaris, Vasileios, I. Kompatsiaris, N V. Boulgouris, and M. G. Strintzis: Real-Time Compressed-Domain Spatiotemporal Segmentation and Ontologies for Video Indexing and Retrieval. IEEE Tr. CSVT, VOL. 14, NO. 5, pp 606- 621, may 2004

8. Werner Bailer, Franz Höller , Alberto Messina, Daniele Airola, Peter Schallauer, Michael Hausenblas: State of the Art of Content Analysis Tools for Video, Audio and Speech. Report, FP6-IST-507336 PrestoSpace Deliverable D15.3 MDS3, 10/03/2005.

9. Jing-Ying Chang, Chung-Jr Lian, Liang-Gee Chen: Architecture and Analysis of Color Structure and Scalable Color Descriptor for Real-Time Video Indexing and Retrieval. Advances in Multimedia Information Processing - PCM 2004, LNCS 3332, pp. 130-137, 2004.

10. W.Stechele: Video Processing using Reconfigurable Hardware Acceleration for Driver Assistance. Workshop on Future Trends in Automotive Electronics and Tool Integration at DATE 2006, Munich, March 6-10, 2006.

11. Jae-Ho Lee, Gwang-Gook Lee and Whoi-Yul Kim: Automatic Video Summarizing Tool using MPEG-7 Descriptors for personal Video Recorder. IEEE Transactions on Consumer Electronics, Vol. 49, No. 3, AUGUST 2003.

12. Messing, D.S, van Beek. P, Errico. J.H., "The MPEG-7 colour structure descriptor: image description using colour and local spatial information", International Conference on Image Processing, Thessaloniki, Greece, 2001, ISBN: 0-7803-6725-1.

13. ISO/IEC JTC1/SC29/WG11, Doc N3321b, ISO/IEC JTC1/SC29/WG11, Doc N3321b: MPEG-7 Visual part of experimentation Model V 5.0. March 2000.

14. ISO/IEC JTC1/SC29/WG11, Doc N3913: Study of CD 15938-3 MPEG-7 Multimedia Content Description Interface – Part 3 Visual. ISO/IEC January 2001 (Pisa).

15. Messing, D.S, van Beek. P, Errico. J.H., "The MPEG-7 colour structure descriptor: image description using colour and local spatial information", International Conference on Image Processing, Thessaloniki, Greece, 2001, ISBN: 0-7803-6725-1.

16. Alan Smeaton, Paul Over: Shot Boundary Detection Task Overview. TRECVID-2006

17. Horst Eidenberger. Avideo Browsing Application Based on Visual MPEG7 Descriptors and Self-Organising Maps. International Journal of Fuzzy Systems, Vol. 6, No. 3, September 2004.

18. Ph. Joly, "les Effets des Effets de transition", support de séminaire, GT3 et GT10 du GDR ISIS, ENST- Paris, Avril 1998.