Architecture and Analysis of Color Structure and Scalable Color Descriptor for Real-Time Video Indexing and Retrieval

Jing-Ying Chang, Chung-Jr Lian, Liang-Gee Chen

Abstract — *Color structure descriptor (CSD) and scalable color descriptor (SCD) provide satisfactory and scalable image indexing and retrieval results among other color-based descriptors in MPEG-7. The superiority of CSD comes from the consideration of space distribution of pixel colors and SCD provides scalability on histogram by a Haar filter. In this paper, we proposed the first hardware architecture that combines CSD and SCD by resource sharing, and it can generate descriptions with frame size 256x256 and 30 frames per second (fps). This architecture provides about 12 times speed-up than running on a 2.54 GHz microprocessor platform to achieve real-time applications like assisting fast algorithm of motion estimation in video coding system and circumstance change detection in surveillance system.*[1]

Index Terms — Hardware architecture, indexing and retrieval, MPEG-7 application, MPEG-7 descriptor.

I. Introduction

With mature digital video technology, inexpensive camcorders gradually enter our life. Original purpose of MPEG-7 is to provide a powerful search engine which helps people easily find what they are looking for. Several MPEG-7 toolkits integrate useful functionalities for categorizing and organizing their personal collection. However, some related research showed that most people only categorize their albums at semantic level, and the recognition technique nowadays is still not able to meet this kind of demand [1]. MPEG-7 descriptors are good tools for indexing and retrieval but should not be limited to them. MPEG-7 descriptors can be creatively extended and linked to applications such as rate control in real-time video coding and movement detection in surveillance systems. In these applications, computational complexity of the real-time implementation for these descriptors will not be a trivial issue.

With statistics derived from MPEG-7 descriptors, good indication of image and video properties can provide referable adjustment parameters for video pre-processing like auto white balance, RGB gains tuning, saturation control, auto contrast, and edge enhancement. In video coding, it can assist fast algorithm of motion estimation, rate control policy, probability distribution model of entropy coding, and so on. A recent research showed that edge histogram descriptor and SCD are applied to segmentation for content-based video coding [2], [3]. When we use them in surveillance system, the system can notice police to keep an eye on unusual behavior by analyzing object trajectory. Face descriptor can also provide auto identification of uncertified people in certain degree.

MPEG-7 visual descriptors record statistics of images and video sequences in color, texture, shape of objects, and motion. Because the variety of possible applications, we first take implementation of color descriptors as our start point. Color is one of important visual attributes for human vision and image processing. It is also an expressive visual feature in image and video retrieval. In MPEG-7, six descriptors are selected to record color statistics of images and video. Among them, CSD and SCD provide better image indexing and retrieval results [4]. In this paper, we focus on the architecture and analysis of resource sharing of CSD and SCD.

Although the concepts of CSD and SCD are different from each other, they have similar color transformation, histogram accumulation, and non-linear quantization. These similarities make it possible to combine these two descriptors.

The challenge to realize CSD part for real-time video system is that each pixel in a frame needs to be scanned 64 times. The vast data bandwidth and then excessive operating frequency make CSD impossible for real multimedia systems. Analysis of the trade-off between input bandwidth and local buffer size is the first issue needed to be evaluated. Then, the

TABLE I
MIPS AND MEMORY BANDWIDTH OF CSD GENERATOR

Operation	1 fps		30 fps	
	Number of instructions (MIPS)	Memory bandwidth (MBytes)	Number of instructions (MIPS)	Memory bandwidth (MBytes)
HMMD	5.625	3.585	168.750	107.550
Accumulation	143.657	202.456	4309.710	6073.680
Quantization	0.051	0.001	1.517	0.039
Other	0.990	0.697	29.713	20.901
Total	150.323	206.739	4509.690	6202.170

4.5 giga instructions per second (GIPS) and 6.2 GB/s of memory bandwidth is the reason why CSD is not suitable for real-time application on software platform.

[1] Jing-Ying Chang is with Graduate Institute of Electrical Engineering, College of Electrical Engineering and Computer, National Taiwan University. Taipei, Taiwan. (e-mail: jychang@video.ee.ntu.edu.tw)

Chung-Jr Lian is with Graduate Institute of Electrical Engineering, College of Electrical Engineering and Computer, National Taiwan University. Taipei, Taiwan. (e-mail: cjlian@video.ee.ntu.edu.tw)

Liang-Gee Chen is with Graduate Institute of Electrical Engineering, College of Electrical Engineering and Computer, National Taiwan University. Taipei, Taiwan. (e-mail: lgchen@cc.ee.ntu.edu.tw)

0-7803-8526-8/04/$20.00 ©2004 IEEE.

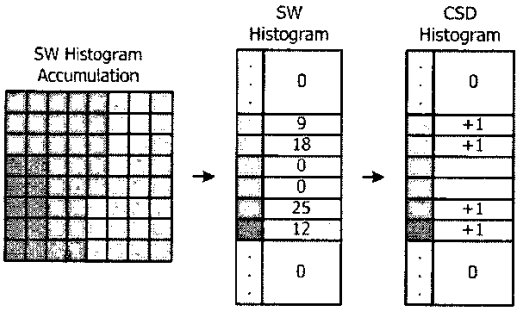

Fig. 1. SW histogram accumulation. Bin values of the colors existing in the SW only increment by one even the color appears several times.

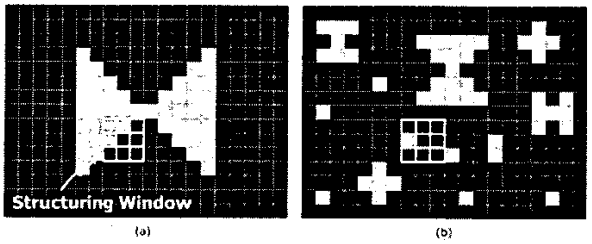

Fig. 2. Two images have the same traditional histogram, but (b) has much more gray components in CSD description.

index algorithm of the color appearance in one structuring window (SW) has to be considered carefully to lower operating frequency. Realizing SCD part is not difficult, so we pay attention to providing a suitable solution for CSD and how to share same resource with two descriptors.

Operational analysis of software simulation for CSD is shown in Table I. "Accumulation" comprises related operations of moving SW and CSD histogram accumulation. For a video sequence with frame size 256×256 and 30 fps, 4.5 giga instructions per second (GIPS) and 6 giga bytes per second (GB/s) of memory bandwidth are needed. Such computational cost is the reason why CSD can not be applied to real-time products without a hardware accelerator. And there is no good solution at present.

In this paper, we first give a brief introduction about the algorithms of CSD and SCD. Next we show the block diagram of CSD and SCD and the similarity between them in section III. Section IV describes architecture of color appearance indexing in CSD. The design of Haar transform and resource sharing issue of SCD are discussed in section V. Section VI shows the experimental result and section VII concludes the remarks.

II. ALGORITHMS OF CSD AND SCD

A. Color Structure Descriptor

CSD represents an image by color accumulation and the local spatial distribution of colors. The procedure of CSD

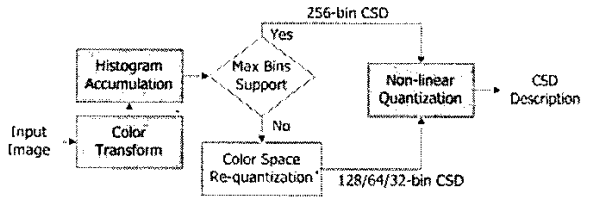

Fig. 3. CSD extraction flow

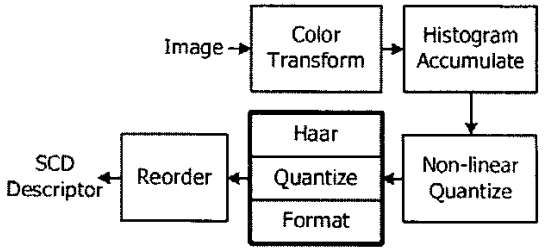

Fig. 4. Block Diagram of SCD. The histogram passes a Haar filter and become energy centralized. SCD description is transmitted from low band to high band.

histogram uses a moving 8x8 SW, which shifts one row or one column at a time, to observe which colors are present in it, and then updates those color bins by only adding one, no matter how many same color pixels exist [5]. This procedure is shown in Fig. 1. Figure 2 illustrates that two images have identical traditional histograms but different CSD descriptions [6]. Figure 2(b) looks more scattered than Fig. 2(a). This characteristic causes gray pixels to exist in more SWs, and finally reflects on gray bin in CSD description. This advantage lets us easily distinguish those images with dissimilar dispersion.

Figure 3 depicts CSD extraction procedure [7]. Our design chose highest number of bins for more precise CSD description in real-time applications. The top path directs the flow of 256-bin CSD that starts with color transformation from RGB to HMMD. Next step is histogram accumulation which is followed by a decision of number of bins needed. After a nonlinear quantization, CSD description is derived.

B. Scalable Color Descriptor

The characteristic of SCD is similar to traditional color histogram. The difference between them is SCD uses Haar filter on the histogram and express it in frequency domain. This approach provides a scalable description because the length of the description can be varied according to the precision we need. Similar to quantization in frequency domain of image/video codec, SCD reserves more bits for "low frequency" bins, and vice versa.

The SCD extraction flow and block diagram is shown in Fig. 4. After converting RGB to HSV color space, traditional histogram accumulation is applied on the image. The results are transformed into frequency domain by using Haar filter, and the output of the filter is reordered according to the

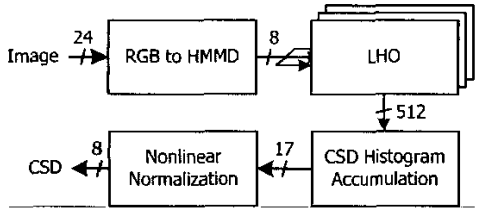

Fig. 5. Block Diagram of CSD. Three parallel LHO indicate which colors are present in each corresponding structuring window.

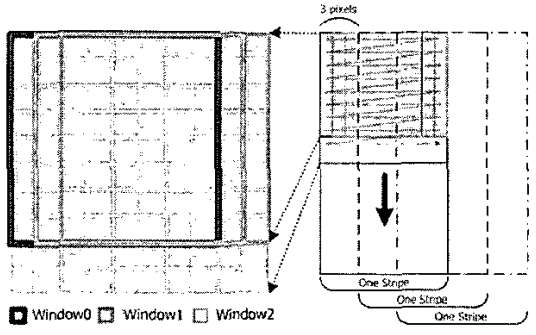

Fig. 6. Pixel scan order of a structuring window. The raster scan order in one stripe provides data reuse merit, which only ten pixels are needed to update three new SWs. After finishing one stripe, index SW colors in next one until all stripes in a frame are visited.

concept of energy centralization. Finally, SCD description is expressed from "low frequency" to "high frequency".

III. Block Diagram of CSD and SCD

CSD block diagram is shown in Fig. 5. After color transform from RGB to HMMD, main part of CSD is the three parallel local histogram observing (LHO) blocks, which are used to indicate which colors are present in each structuring window [8], [9]. After finishing color indicating of three SWs, their results are summed and sent to CSD histogram accumulation. SCD block diagram is illustrated in Fig. 4 and stated in last section.

First similarity between CSD and SCD are the color transforms from RGB to Hue-Max-Min-Difference (HMMD) and RGB to Hue-Saturation-Value (HSV). Evaluation of saturation in HSV is the only overhead. Second similarity is non-linear quantization. Architectures of the two quantization blocks are the same, except for the quantization tables. The other similarity is not the characteristic of two algorithms but the local buffer needed for them. The buffer in each LHO in CSD just fits the need of the buffer of Haar filter in SCD.

IV. Color Appearance Recording in CSD

A. Parallelism Analysis

Specification of our CSD generator is for the video sequence with frame size 256×256 and 30 fps. Operating frequency limitation is targeted at 27 MHz, which is common for most TV systems. This requirement can be achieved by buffering three successive SWs (8×10 pixels). Purpose of the buffer is for data sharing. The scan order is shown in Fig. 6. Pixel values of three SWs are complete updated after discarding top row pixels from last three SWs and reading in ten new bottom pixels in current SWs. After finishing indexing SW colors in one stripe, we start to index SW colors in next stripe. The displacement between adjacent stripes is three pixels.

TABLE II
Relationship between Parallelism and Operating Frequency

Parallelism	Memory Bandwidth (MB/s)	Working Freqency (MHz)
0	357	476
1	46	61
2	26	35
3	19	25
4	16	21

Zero parallelism means no SW is buffered. The minimum requirement to meet target frequency (27 MHz) is three parallelisms.

Parallelism decision is according to the target frequency. Approximately, in the situation of no local buffer of SW, each pixel in every window has to be scanned again even though it has been scanned during the period of operations of last neighboring window. The memory bandwidth is about 357 MB/s and the required operating frequency is 119 MHz. In fact, we assume histogram can be updated once in one cycle to make this chip running at 119 MHz. But according to the problem of single-port SRAM processing speed, it takes four cycles to update one pixel data on average and forces the required operating frequency to 476 MHz. Relationship of parallelism and operating frequency is shown in Table II. Three parallelisms is the final decision to meet the requirement without over design.

B. Color Appearance Recording in LHO

How to record which colors exist in a SW efficiently is another main issue. It is unrealistic to query all pixels at the same time or to query by taking 64 cycles. The method of querying at the same time will make interconnection of decision circuit become very large and inconvenient to handle. The method of querying by taking 64 cycles has to be realized by raising operating frequency. In order to solve the problem, we proposed LHO architecture. LHO contains a SRAM to record color histogram of a SW and a color appearance register bank to indicate which colors exist in the SW according to the values of the color bins.

The main idea of LHO is recording SW histogram to indicate which colors exist in a SW. Along with updating histogram, we observe the value of changing color bin and save this information into color appearance register bank. Nonzero bin means this color belongs to the window. After update, three register banks are summed and sent to CSD histogram accumulation block.

Using SRAM to record histogram of SW is an area efficient method. But histogram updating cycles are directly restricted

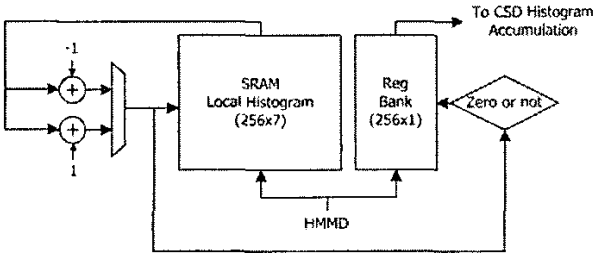

Fig. 7. Structuring window histogram updating architecture. HMMD values from color transform and SW buffer indicate which bin in the SW histogram needed to be added or subtracted by one.

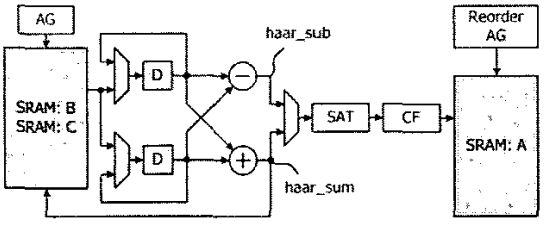

Fig. 8. Haar transform architecture. The two registers (D) store the data as inputs of Haar filter. SAT block performs saturation for output of Haar filter. CF block changes two's complement representation to sign-magnitude format.

by SRAM specification. Single port SRAM provides one read or one write in a cycle. That means, when we get an address from the color which needs to update corresponding color bin, we read the bin value in one cycle, add or subtract the value by 1, and write it back to SRAM in another cycle. With an appropriate design for dual port SRAM, the throughput of updating histogram can achieve one update per cycle at the expense of double SRAM area and power. With power consideration, we choose single port SRAM as buffer of SW histogram. Single port SRAM takes four cycles to refresh histogram for each pixel. Two cycles are for removing accumulation from previous pixel and the others are for addition of incoming pixel. To update three SWs by refreshing ten pixels will take 40 cycles. Figure 7 depicts the LHO architecture.

V. HAAR TRANSFORM AND RESOURCE SHARING IN SCD

A. Haar Transform

Haar transform in SCD works on the color histogram to present the histogram by way of the form of "frequency domain". As the precision that users require, SCD descriptions are sent from low band to high band data. The architecture of Haar transform is shown in Fig. 8. Two bin values are saved in two registers and ready for summation and subtraction of Haar filter. Most of the time, the subtraction data representing high band are transmitted to saturation and format changing block as final output. The last output of Haar filter represents the bin with the lowest frequency and is sent to saturation and format changing block at the last cycle. Three SRAMs in Fig. 8 share local buffer of three LHOs in CSD.

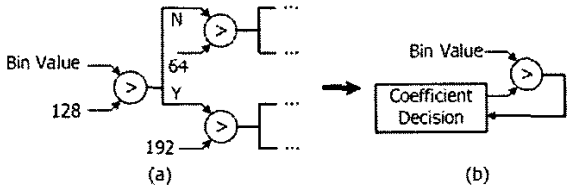

Fig. 9. (a) shows the binary comparison concept. The quantization candidate is compared with central value within the valid range each time. (b) shows we can fold (a) structure in single comparator.

B. Color Transform

As mentioned before, color transformations of CSD and SCD are similar. Hue exists in HMMD and HSV, and occupies greater part of computation in the transformations. The RGB to HMMD transformation formula is listed below.

```
MAX = max(R, G, B);   MIN = min(R, G, B);   DIFF = MAX - MIN;
if(MAX == MIN)                              HUE = 0;
else{
    if((MAX == R) && (G >= B))              HUE = 60 x (G - B) / DIFF;
    else if((MAX == R) && (B > G))          HUE = 360 - 60 x (B - G) / DIFF;
    else if(MAX == G)                       HUE = 120 + 60 x (B - R) / DIFF;
    else                                    HUE = 240 + 60 x (R - G) / DIFF;
}
```
(1)

The most time consuming and area occupied part of (1) is evaluation of hue value. If the divider and the multiplier were directly mapped into hardware, it would require high precision and set a bottleneck of the chip. Since dividend and divisor are 8-bit integers and final output of hue is quantized into 16 categories, two drawbacks just mentioned would be eliminated by building a mapping table for hue evaluation. According to our synthesis results, realizing color transformation by table look-up method can save 36% area and shorten critical path than by fundamental operations. Saturation evaluation in HSV is the only overhead. Similar to establishing look-up table of hue, we also make a look-up table for saturation.

C. Non-linear Quantization

Non-linear quantization of CSD and SCD can be achieved by using same binary comparison method and folding skill. In CSD, after histogram accumulation is finished, non-linear histogram quantization is the final step. Each bin should be quantized into 8-bit via 255 comparisons. With those skills, eight comparisons are needed to quantize one bin. This strategy is shown in Fig. 9. As shown in (a), we compare the bin with center value within valid range each time. Since the latency of non-linear quantization, which is compared with CSD histogram accumulation, is negligible, 255 comparators can be folded into one. With (b) architecture, 2048 (256x8) cycles and one comparator are needed to achieve this work. In SCD, non-linear quantization is applied before Haar transform. Each bin is quantized into 4 bits. With same architecture but different quantization table, 1024 (256x4) cycles are required

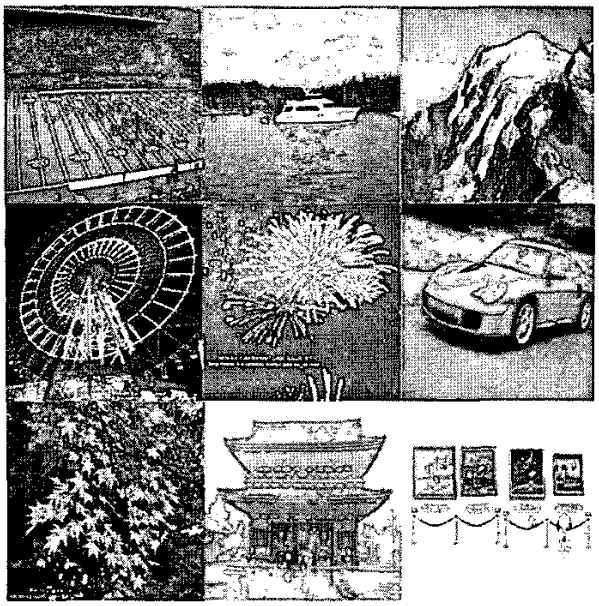

Fig. 10. Sample images of experimental database. The indexing and retrieval database contains 526 images in 78 categories.

to finish the quantization.

VI. Experimental Result

In this section, we show that indexing and retrieval results of CSD and SCD for comparison. The indexing and retrieval database contains 526 images in 78 categories. Those images are collected from Internet and manual categorized. The quantity of images per category varies from 2 to 28. The semantic categories include landscapes, animals, buildings, transportation, night scenes, paintings, cartoons, and so forth. Nine sample images are showed in Fig. 10. Furthermore, for extending the concepts of these descriptors to image and video coding, we replace default color spaces with YCbCr domain and the performance drops slightly.

Here we use a quantitative measure method called query-by-example (QBE) suggested by MPEG-7 [6]. QBE sorts the distances between description vector of query image and those of images contained in a database. Retrieval rank represents the rank at which certain ground-truth image is retrieved. Normalized modified retrieval rank (NMRR) eliminates the influence of number of ground-truth images. Finally, average normalized modified retrieval rank (ANMRR) is the average of NMRR of each query. The smaller ANMRR means the descriptor provides better indexing and retrieval ability. Table III shows the indexing and retrieval results of CSD and SCD with designated and YCbCr color spaces. ANMRR of CSD with HMMD is the lowest as our expectation. And the results of two descriptors with YCbCr are also acceptable. Because of the characteristic of our database and subjective manual categorization, the values in this table are smaller than the experimental results in [6]. These good results imply that we can apply the concepts of these descriptions to the field of image and video coding which chooses YCbCr as default color space.

TABLE III
Indexing and Retrieval Results of CSD and SCD

Descriptor	Color space	ANMRR
CSD	HMMD	0.00105097
CSD	YCbCr	0.00360790
SCD	HSV	0.00165656
SCD	YCbCr	0.00428604

Smaller ANMRR value means better retrieval result.

VII. Conclusion

In this paper, we provide the vision of future MPEG-7 descriptor applications for not only indexing and retrieval, but also for real-time multimedia applications. First analysis of dedicated hardware architecture design for combination of CSD and SCD descriptors, which can generate CSD and SCD descriptions together with frame size 256×256 and 30 fps, is also proposed. This design provides about 12 times speed-up than running on a 2.54 GHz microprocessor platform to achieve real-time applications. Detailed design explorations of the hardware implementation, and practical reference data of prototype is valuable for future researchers.

References

[1] Kenneth R. Wood Kerry Rodden, "How do people manage their digital photographs," in Proceedings of the conference on Human factors in computing systems, ACM Press, 2003, pp. 409–416.

[2] Patrick Ndjiki-Nya and Oleg Novychny, "A MPEG-7-aided segmentation tool for content-based video coding," in Proc. International Symposium on Circuits and Systems 2004,May 2004, pp. III – 849–852.

[3] Patrick Ndjiki-Nya, Oleg Novychny, and Thomas Wiegand, "Merging MPEG-7 descriptors for image content analysis," in Proc. International Conference on Acoustics, Speech, and Signal Processing 2004, pp. III – 453–456.

[4] T.Ojala, M.Aittola, and E.Matinmikko, "Empirical evaluation of MPEG-7 XM color descriptors in content-based retrieval of semantic image categories," in Proc. International Conference on Pattern Recognition 2002, August 2002, vol. 2, pp. 1021–1024.

[5] R.J. Qian, P.J.L. Van Beek, and M.I. Sezan, "Image retrieval using blob histograms," in Proc. International Conference on Multimedia and Expo 2000, August 2000, vol. 1, pp. 125–128.

[6] B.S. Manjunath, Philippe Salembier, and Thomas Sikora, Introduction to MPEG-7, pp. 204–208, JOHN WILEY and SONS, LED, 2002.

[7] ISO/IEC JTC 1/SC 29/WG11 N4062, Text of ISO/IEC 15938-3/FCD Information technology - Multimedia content description interface - Part 3 Visual, pp. 47–52, March 2001.

[8] Jing-Ying Chang, Hung-Chi Fang, Yen-Wei Huang, and Liang-Gee Chen, "Architecture of MPEG-7 color structure description generator for real-time video applications," in Proc. International Conference on image processing 2004, October 2004, submitted for publication.

[9] Jing-Ying Chang, Chung-Jr Lian, Hung-Chi Fang, and Liang-Gee Chen, "Architecture and analysis of color structure descriptor for real-time video indexing and retrieval," in Proc. Pacific-Rim Conference on Multimedia 2004, December 2004, submitted for publication.