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

Jing-Ying Chang, Chung-Jr Lian, Hung-Chi Fang, and Liang-Gee Chen

DSP/IC Design Lab.,
Graduate Institute of Electronics Engineering,
National Taiwan University, Taipei, Taiwan
{jychang, cjlian, honchi, lgchen}@video.ee.ntu.edu.tw
http://video.ee.ntu.edu.tw

Abstract. Color structure descriptor (CSD) provides satisfactory image indexing and retrieval results among other color-based descriptors in MPEG-7. The superiority comes from the consideration of space distribution of pixel colors. In this paper, we proposed the first CSD hardware architecture which can generate CSD description with frame size 256×256 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 rate control in video coding system and circumstance change detection in surveillance system.

1 Introduction

With mature digital video technology, inexpensive camcorders gradually enter our life. More and more multimedia are produced and shared among the world. Original intention of MPEG-7 is to provide a powerful search engine which helps people easily find what they are looking for. Some MPEG-7 toolkits further integrate useful functionalities for categorizing and organizing their personal collection. However, some related research [1] showed that most people only categorize their albums at semantic level, but the recognition technique nowadays is still not able to meet this kind of demand. 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 loads 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 preprocessing 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

K. Aizawa, Y. Nakamura, and S. Satoh (Eds.): PCM 2004, LNCS 3332, pp. 130–137, 2004.

coding, and so on. 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. Color descriptions usually are irrelevant to viewing angle, translation and rotation. This advantage possesses good resistance to undesired shacking of camera. In MPEG-7, six descriptors are selected to record color statistics of images and video. Among them, CSD provides best image indexing and retrieval results[2]. The superiority comes from that CSD considers space distribution of pixel colors by recording appearance of each color in every structuring element window in its histogram [3]. In this paper, we focus on the architecture and analysis of CSD.

The challenge to realize CSD hardware accelerator 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 index algorithm of the color appearance in one structuring window (SW) has to be considered carefully to lower operating frequency. Along with exploring suitable solutions, hardware extensibility should not be left behind. It is worth to integrate with other descriptors with small overhead.

Operational analysis of software simulation is shown in Table 1. "Accumulation" comprises related operations of moving SW and CSD histogram accumulation. For a video sequence with frame size 256×256, 30 fps, 4.5 giga instructions per second (GIPS) and 6 giga bytes per second (GB/s) of memory bandwidth are required in one second. 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 describe briefly the algorithm of CSD in section 2. Before going into implementation details of each functional block, hardware issue and operational parallelism are discussed in section 3.1 and then each block design. Section 4 shows the experimental result and summarizes the chip specification. Section 5 is dedicated to concluding remarks and future research.

2 Color Structure Descriptor

CSD represents an image by color accumulation and the local spatial distribution of colors. The procedure of CSD histogram uses a 8×8 SW to observe which colors are presented in it, and then updates those color bins by only adding one, no matter how many same color pixels exist. Figure 1 shows that two images have different CSD description with the same traditional histogram[4]. Right

Table 1. MIPS and memory bandwidth of CSD generator.

	1 fps		30 fps	
Operation	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
Others	0.990	0.697	29.713	20.901
Total	150.323	206.739	4509.690	6202.170

image looks more scattered than left one. Such situation causes gray pixels exist in more SWs and reflects on gray bin in CSD description. This advantage let us easily distinguish those images with similar dispersion.

Figure 2 depicts CSD extraction procedure [5]. 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. It 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.

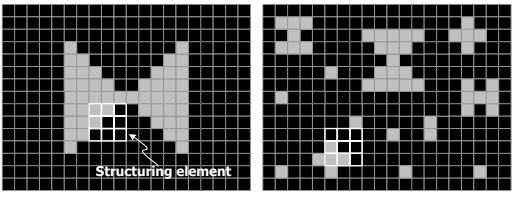

Fig. 1. Two images have the same traditional histogram, but right one has much more gray components in CSD description.

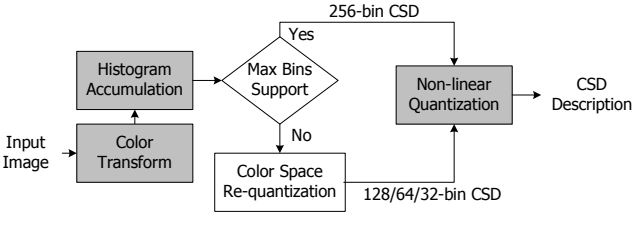

Fig. 2. CSD extraction flow.

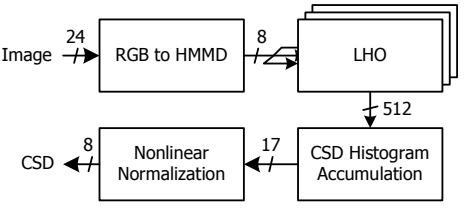

Fig. 3. Block diagram of CSD architecture.

3 Computational Complexity and Proposed Architecture

As described in Section 1, we focus on real-time applications of MPEG-7 like video coding assistance and surveillance systems. Besides, generated CSD descriptions still can be used for search of multimedia contents. And for supporting comparison with descriptions generated by other tools, 256 levels of color quantization is adopted for downscale comparison.

Since a sub-sample factor is defined in the standard for large images, we choose 256×256 as input image size. The sub-sample factor, K, is defined as $K = \max\{1, 2^{\lfloor \log_2 \sqrt{W \cdot H} - 7.5 \rfloor}\}$, where W and H are the width and height of image. For example, $K = 2$ implies an image is sub-sampled by 2 horizontally and vertically. Note that the SW size is always 8×8.

Our CSD block diagram is shown in Fig. 3. After color transformation, pixels are sent to corresponding local histogram observing (LHO) blocks and index colors that exist in these windows. Summation of outputs of three LHO blocks indicates how many windows does each color belong in. Then, the summation updates CSD histogram. Finally, CSD description is obtained via non-linear quantizing the completed histogram.

3.1 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. 4. 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.

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 megabytes/sec (MB/s) and the required operating frequency is 119 MHz. In fact, we assume

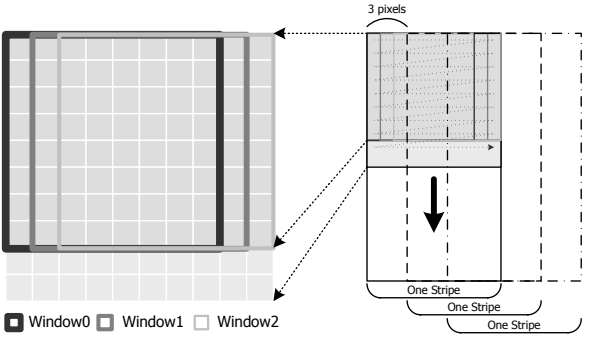

Fig. 4. Pixel scan order of three structuring windows.

Table 2. Relationship between parallelism and operating frequency. Zero parallelism means no SW is buffered. The minimum requirement to meet target frequency (27 MHz) is three parallelism.

Parallelism	MB/s	MHz
0	357	476
1	46	61
2	26	35
3	19	25
4	16	21

histogram can be updated once in one cycle to make this chip running at 119 MHz. But according to the problem described in section 3.2, 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 2. Three parallelism is the final decision to meet the requirement without over design.

3.2 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 a 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

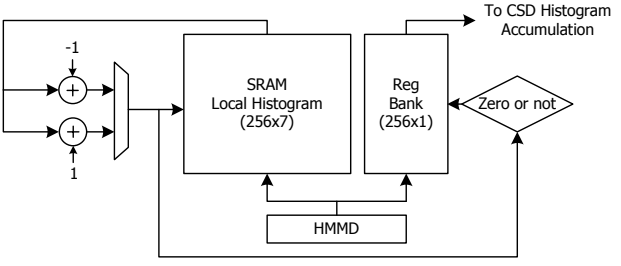

Fig. 5. Structuring window histogram updating architecture.

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 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 5 shows the LHO architecture.

3.3 CSD Histogram Accumulation

According to cycle analysis in section 3.2, there are 40 cycles to complete 256-bin CSD histogram accumulation. If we store CSD histogram in SRAM and wish to refresh it in time, that means we have to update 16 color bins in two cycles, improper bit-width and number of addresses of SRAM will cause this SRAM occupies large area and waste much power. Here we divide this SRAM into four to lower the unreasonable bit-width. The bit-width is equal to four color bins. Each SRAM is 16×64 bits.

3.4 Non-linear Quantization

After CSD histogram accumulation is finished, non-linear histogram quantization is the final step. Each bin should be quantized into 8-bit via 255 comparisons. With binary comparison method and folding skill, eight comparisons are needed to quantize one bin. This strategy is shown in Fig. 6. As shown in (a), we compare the bin with center value of valid range each time. Since the latency of non-linear quantization, which is compared with CSD histogram accumulation,

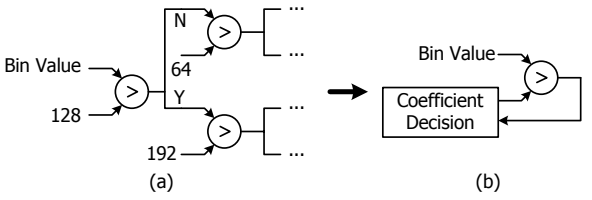

Fig. 6. Folding skill on non-linear quantization.

Table 3. Indexing and retrieval result

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

Table 4. Chip specification

Technology	UMC 0.18 μm CMOS 1P6M
Core size	$1.36950 \times 1.36584\ mm^2$
Gate count	49865
On-chip single port SRAM	11136 Bits
Max frequency	31.25 MHz
Operating frequency	27 MHz
Processing speed	256×256, 30 fps@ 27 MHz
Power Consumption	39.53 mW@ 27 MHz, 1.8 V

is negligible, 255 comparators can be folded into one. With (b) architecture, 2048 (256×8) cycles and one comparator are needed to achieve this work.

4 Experimental Result

Our indexing and retrieval database contains 526 images in 78 categories. Those images are collected from Internet and manual categorized. 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 [4]. QBE sorts the distances between description vector of query image and those of images contained in a database. The smaller average normalized modified retrieval rank (ANMRR) means the descriptor provides better indexing and retrieval ability.

Table 3 shows the indexing and retrieval results of CSD and scalable color descriptor (SCD) with designated and YCbCr color spaces. SCD listed here is for comparison. The results with YCbCr are also acceptable and imply that we can apply the concepts to the field of image and video coding which chooses YCbCr as default color space.

This first proposed CSD hardware architecture for realtime applications can generate CSD description with frame size 256×256 @ 30 fps. Detailed specification is shown in Table 4.

5 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 MPEG-7 CSD descriptor is also proposed. Detailed design explorations of the hardware implementation, and practical reference data of prototype is valuable for future researchers. The integration with SCD by sharing much existed resource is ongoing. In the future, descriptors with similar architecture can be integrated into this design.

References

1. Kerry Rodden, Kenneth R. Wood: How Do People Manage Their Digital Photographs. Proceedings of the conference on Human factors in computing systems, ACM Press, New York, NY, USA (2003) 409–416
2. Ojala, T. and Aittola, M. and Matinmikko, E.: Empirical evaluation of MPEG-7 XM color descriptors in content-based retrieval of semantic image categories. IEEE International Conference on Pattern Recognition, 2002, Vol. 2 (August 2002) 1021–1024
3. Qian, R.J. and Van Beek, P.J.L. and Sezan, M.I.: Image retrieval using blob histograms. IEEE International Conference on Multimedia and Expo, 2000, Vol. 1 (August 2000) 125–128
4. B.S. Manjunath and Philippe Salembier and Thomas Sikora: Introduction to MPEG-7. JOHN WILEY and SONS (2002) 204–208
5. Leszek Cieplinski and Munchurl Kim and Jens-Rainer Ohm and Mark Pickering and Akio Yamada: Text of ISO/IEC 15938-3/FCD Information technology - Multimedia content description interface - Part 3 Visual, ISO/IEC JTC 1/SC 29/WG11 N4062 (March 2001) 47–52