International Journal of Computer Sciences and Engineering Systems
Vol. 3 No. 1 (January-June, 2018)

A High-Performance Object-Based MPEG-4 Coding Technique For Video Surveillance

Shih-Chang HSIA, Szu-Hong WANG and Chung-Ying HUNG

Department of Computer and Communication Engineering, National Kaohsiung First University of Science and Technology, Kaohsiung 824, Taiwan, R.O.C.*Email: hsia@ccms.nkfust.edu.tw*

Manuscript received October 23, 2017, Manuscript revised December 15, 2017

Abstract: *In this paper, we present the content-based video coding, MPEG-4 for video surveillance system using software programming. Due to high computational complexity in MPEG-4 system, we proposed some fast MPEG-4 algorithms for MPEG-4 encoding in order to real-time implementation. The fast algorithms include shape coding and object tracking. The shape coding employs the differential code consisting of entropy code to reduce the coding rate for intra-frames. The adaptive motion estimation is used for inter-frames. The object tracking uses the relativity of shape and content information to reduce computational complexity and time of motion estimation. By combination of these fast algorithms, under the CPU using Intel Pentium-4 3.2GHz, the encoder can realize 15~25 frames pre second in off-line. As for real-time camera sampling and encoding, the frame rate can achieve about 2~5 frames.*

Keywords: **Segmentation, Video Object, Shape Information, MPEG-4**

1. INTRODUCTION

The growing demands for transmission and storage of digital video has researched for the development of video coding standards, such as MPEG-1/2/4, H261, H263, and H.264 [1-3] that are designed for different applications. The goal of the MPEG-2 standard is to apply the applications that allow visual compression of DVD and digital TV. In the HDTV with wide-bandwidth transmission environment, the MPEG-2 provides some elementary feature such as simple coding, fast encoding/decoding. The video is compressed under the demand for the large-size resolution, the MPEG-2 supply realization of software or hardware is easier and the cost is cheap, so well accepted by market. MPEG-4 system can use object-based coding to increase the coding efficiency.

In this paper, we present some algorithms such as shape coding and motion object tracking for MPEG-4 system, to reduce the computational complexity, for applying on low-cost real-time surveillance systems. This paper is organized as follows. The fast algorithms are proposed in Section II. The fast algorithms are integrated to MPEG-4 and simulated in section III. Concluding remarks are outlined in Section IV.

2. FAST ALGORITHMS FOR MEPG-4 CODING

2.1 Shape Coding

In the shape coding, we employed the differential chain coding (DCC) to implement our system, according to the general chain coding with differential concept to improve the compression rate. In basic chain coding methods, the object boundaries are represented by a closed contour. Derivation of the contour from the binary alpha plane is similar to the conventional edge direction in image processing. After the contour is obtained, starting from one pixel, the coding scheme moves along the contour in one of the eight or the four directions. The chain coding terminates when all the contour pixels are coded.

To explain the procedure of the differential coding, we first define some parameters as illustrated in Fig. 1. "d_in" means the previous chain coding, which is define as the (incident) direction of the previous pixel entering the current pixel. In other words, this is the edge direction defined by the previous and the current pixels. Similarly, "d_out" means the next chain code, which is the edge direction between the current pixel and the next one along the contour. "d_dif" is then defined as the differential chain code between the "d_in" and the "d_out".

According to the simulation statistics and observation, most of the edges either preserve the direction or simply change the direction by only one pixel left or right, i.e., direction 7 and 1. Taking the advantage of the correlation between the two successive chain codes, the differential chain code is defined as follows

$$d\_dif = (d\_out - d\_in) \bmod 8 \tag{1}$$

To reduce the coding bits, the Huffman coding is adopted to the code for the eight possible differential chain codes. Table.1 lists the results from the comparison with bitmap coding (QTSC & CAE) and quad-tree (QT) based coding.

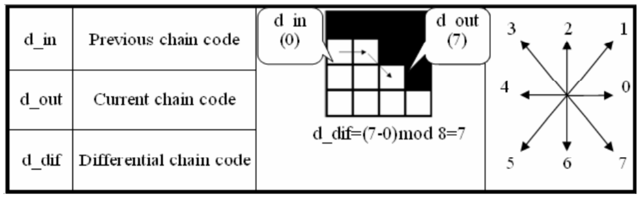


Figure 1: The Parameters for the Differential Chain Coding.

Table 1
The Coding Bits for Shape Coding Algorithms

	Bitmap based		*QT based*		*Chain coding*		
	CAE [4]	*MMR [5]*	*QTSC [6]*	*IQTSC [6]*	*Chain [7]*	*DCC [7]*	*Proposed*
Weather	211	218	834	180	288	153	131
TV	279	286	467	235	367	186	161
Irene	244	291	426	236	357	193	164
Make	549	1238	1073	372	635	308	239

From the Table 1, the proposed chain coding outperforms all other schemes, and the rank is the top level in the coding efficiency. For edge characteristic of the fourth video sequence "Make" belong to the related continuity. Such a case processed under the procedure of DCC can increase the coding efficiency.

2.2 Fast Motion Tracking Algorithm

In order to obtain the efficient track method of moving object, first we employ the binary shape tracking to find possible the moving direction and offset of object. Thus, the detailed offset location of content can be found according to the shape information. There are two main parts in the moving object tracking for inter-frames coding, one is the binary shape motion tracking and the other is the motion search of gray content.

Binary shape motion tracking. We can employ the moving boundary of binary shape to find the rough motion vector. In the rough motion estimation, we can use three rough directions to describe this algorithm as follow. Fig. 2(a) shows that the object has the vertical boundary with horizontal moving. Fig. 2(b) shows that the object has the horizontal boundary with vertical motion.

With XOR operator, we can easily find the motion vector between the previous frame shape and the current one. When the motion is zero, the XOR result is zero too. The motion vector is shown the white pixel of boundary. When motion is high, the number of white pixels becomes high.

Fig. 2(c) and (d) shows the diagonal edge shape with 45 and 135 degree. The XOR result shows motion vector in the number of white pixels in the different side.

When the rough motion vector is found, the fine vector is computed by starting from the rough motion vector. We first check 9 points vector, when the center point is from

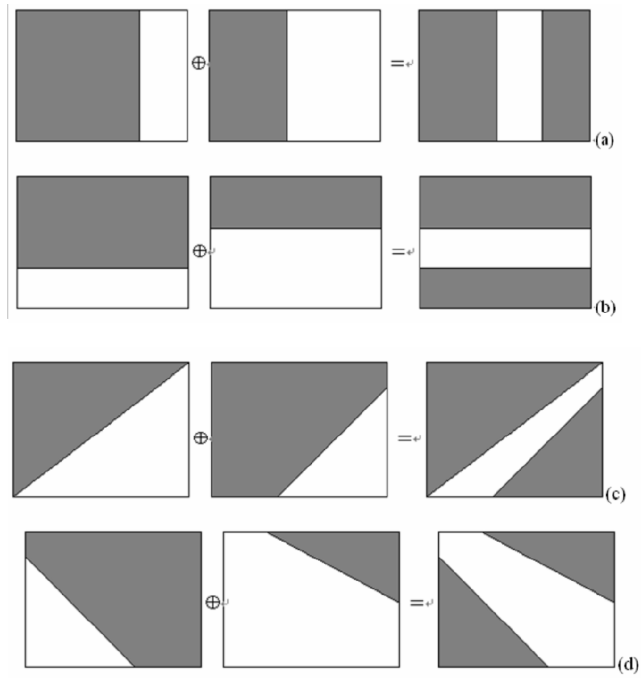


Figure 2: (a) and (b) vertical and horizontal shape motion. (c) and (d) 45 and 135 degree shape motion

the rough vector. When minimum SAD is at the central point, the search is stop, and one can output the vector. When the minimum SAD is at the boundary point, we enter to the 2nd search, as shown in Fig. 3. The search may enter the 3rd route when the minimum SAD point is not located at the boundary after the 1 and 2 searching flows. However, the search will be stopped at the 3rd step, which does not care the final vector location in this case. The number of searching point is from 9 to 21, where the complexity can be efficiently reduced.

The simulation of image size is CIF 352×288 binary file, and compared with full search(FS) and three-step search (TSS), with the search range -16~+16. From Fig. 4(a) shows the mean square error (MSE). Our searching error is less than the three step method. However, our complexity is lower than the other algorithms in. Fig. 4(b). The result of FS has better quality but it costs many of search computation to cause longer processing time, so is not suitable for the real-time coding. When our algorithm is compared with FS, MSE increases very little but our complexity can be greatly saved. Compare to TSS, the calculation is only the half of TSS, and the MSE is less.

Fast Object Motion Tracking: We adopt the search result of binary shape for the base of object tracking to improve the search efficiency. The red square of Fig. 5 is the edge block. These blocks can be referred with corresponding to the binary shape block. The motion estimation can perform the full search with small searching range to cover the enough accuracy. In simulations, the search range for object block is ±4 that this is enough to cover motion feature.

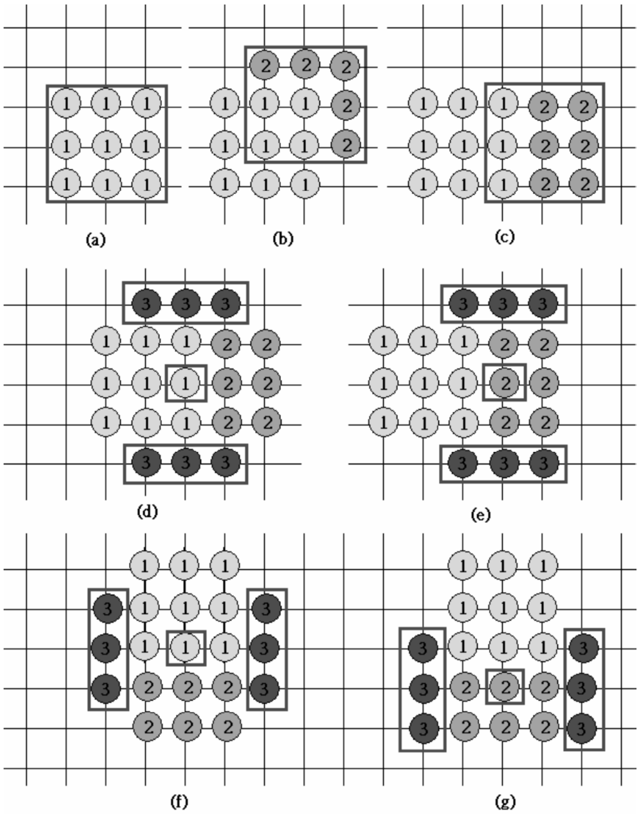
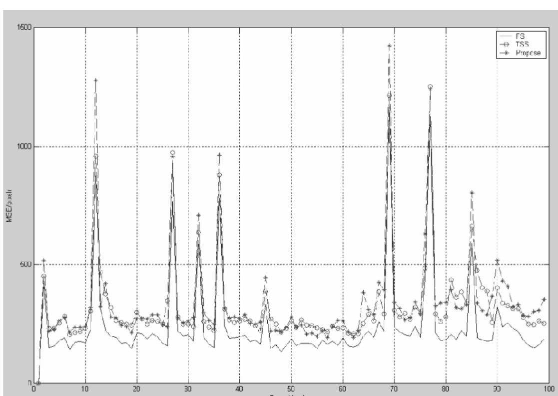
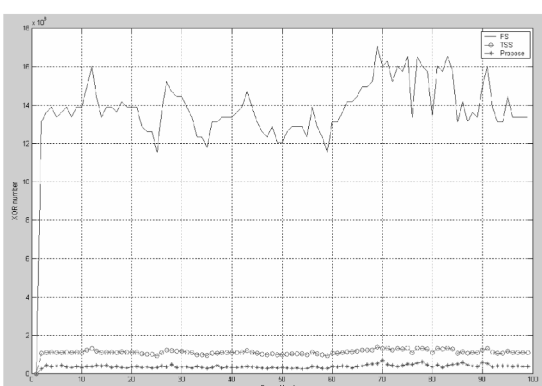


Figure 3: Fine Motion Search Pattern



(a)



(b)

Figure 4: (a) MSE Results (b)The Number of Pixel Comparsion.

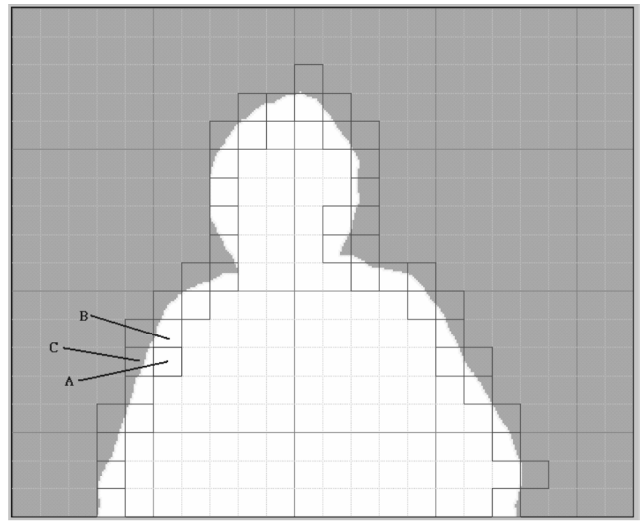


Figure 5: The Fast Object Tracking with Boundary Blocks.

Table 2
Performance Comparisons with Various Methods

	Total time (s)	*Number of time of add*	*Average of PSNR*
FS	102.061	4,193,377k	28.209
TTS	8.372	339,512k	27.8086
Propose	4.732	187,417k	27.6394

Another case is deal with the white block, such as ¡§A¡¨ block in Fig. 5, which is the inside block for the object. The reference motion vector for block A from the average for motion vectors of the block B and the block C. Thus the boundary block must be estimated before search block A vector. The center vector of block A adopts the average vector as the base and to take the full search with range of ±4. The other inside-blocks are processed with the same scheme. Table 2 lists the results of motion estimation. The processing speed of our algorithm can improve about 2 times compared to TTS with slightly quality degradation.

3. FAST ALGORITHMS INTEGRATION FOR MEPG-4 CODING

The red coarse-line parts of Fig. 6 are the fast algorithms. There are two main phases in the encoder, the top part is the shape coding and the bottom part is for the context coding. For the shape phase coding, first the process information of binary shape is decided via a selector to carry out the shape coding or motion estimation. We output the shape information through the multiplexer when the selector used the shape coding. On the other case, for inter shape coding, the motion vector are outputted to the multiplexer. After the shape coding and motion estimation have been processed, the information can be used to reconstruct the shape and store it to memory. The reconstructive shape is used to dealing with the next shape information.

After the shape phase has been processed, following the object motion tracking is performed. First, there will perform gray content motion search according to results of binary shape motion. Then, we take motion estimation to find the motion vector and offset, and to output the motion vector information of the object content to the multiplexer. The offset image is subtracted from the original image to produce a residual, and the residual using the texture coding with fast DCT[8] and entropy. Then the residual DCT information for the content data is sent to the multiplexer. At the same time, the content information and the residual are added to produce the reconstructed image, and stored it to the memory and it will be used to code the next image.

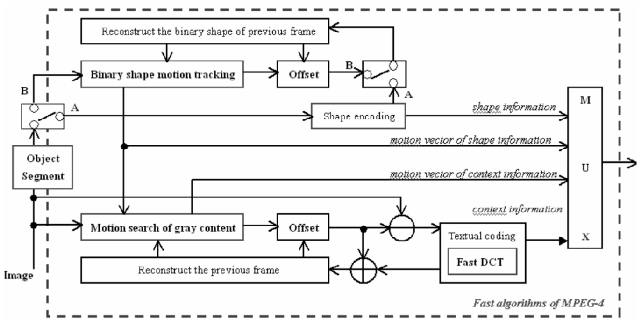


Figure 6: Integration the System of MPEG-4 Coding

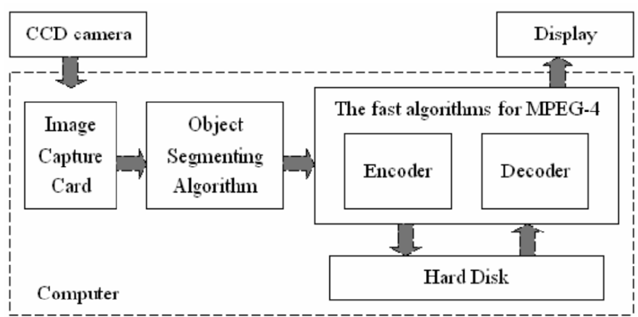


Figure 7: The Architecture Diagram for Simulation Video Surveillance System

Fig. 7 shows the system block to simulate the video surveillance system. The image capture card is used to sample the video to the computer through PCI bus. The video is auto-segmented to separate the object and background. The background is coded by intra-coding with DCT and VLC core. For the texture coding, the fast shape coding and object coding is used. The coding bit is saved to the hard disk in real-time operation. After a period of coding, we can play the recorded video from the MPEG-4 decoder, and the video is reconstructed and output to monitor.

In the off-line simulations, we take three kinds of the sequence length and three kinds of the object size. The parameters used the compressing rate (CR), PSNR, and FPS (frame per second). The simulation specification is as follows. The image size is 352x288, while the sequence lengths are 10 second (approximately 40 pictures), 100 second (approximately 614 pictures), and 1000 second (approximately 6320 pictures), while the object size are large (approximately 200 macro-blocks, account for 50% the picture), middle (approximately 100 macro-blocks, account for 25% the picture), and small (approximately 40 macro-blocks, account for 10% the picture).

Table 3
(a) The off-line Simulation for Various Object Size
(b) The Real-time Implementation with C-programming for Various Object Size*

	Large object (50%)			*Middle object (25%)*			*Small Object (10%)*		
	CR	*PSNR*	*FPS*	*CR*	*PSNR*	*FPS*	*CR*	*PSNR*	*FPS*
10	67.10	34.22	10.00	87.09	33.95	9.75	172.89	34.32	15.67
100	100.12	32.61	9.90	154.67	33.85	12.28	350.54	34.34	21.17
1000	115.43	30.21	11.59	287.60	33.68	16.59	640.69	31.37	24.76

(a)

	Large object (50%)			*Middle object (25%)*			*Small Object (10%)*		
	CR	*PSNR*	*FPS*	*CR*	*PSNR*	*FPS*	*CR*	*PSNR*	*FPS*
10	60.63	34.10	3.9	105.32	33.84	4.2	151.32	34.80	4.6
100	76.93	32.08	4.02	208.48	33.46	4.46	324.34	34.06	4.75

(b)

where CPU used Pentium-43.2 Ghz.

Due to object-based coding, the frame rate can achieve higher when the object size is small. In the real surveillance system, if the recoding time is longer, we have more the static backgrounds. Moreover, the texture content is small difference in the same object between the inter frames. This lead to high CR. Table 3(a) shows the off-line simulations. If the object is smaller under long recording time, the CR, PSNR, and FPS values are relative high. Table 3(b) shows the simulation results of real-time operation with C-programming. The sampling, segmenting and coding are performed at the same time. The processing speed is limited by the image capture card and the accessing time of hard disk, where the FPS will drops to 4~5.

4. CONCLUSIONS

This paper proposed some fast algorithms of MPEG-4 suitable for surveillance system. In order to achieve real-time processing, the most significant compression ability use the MPEG-4 technique to reduce the necessary capacity while keeping the image quality in the acceptable range. The auto-segmented object technique is used to cut the objects correctly in real-time. The fast shape coding and motion tracking are employed to reduce the computation time of shape coding and motion estimation. Finally, we also employed the run-length coding to reduce coding length and increase compression rate. In off-line operation, we pre-store the sampling video in disk. The processing speed for MPEG-4 coding can achieve about 15~25 frames per sec, In real-time coding with sampling case, the frame rate is about 4~5. According to the results of simulation, these algorithms can maintain the PSNR above 30dB in quality, and the compression rate will keep on 60~600 times dependent on the object size. With High-compression, high quality and low complexity, the proposed coding techniques can be suitable for low-cost video surveillance system.

REFERENCES

[1] Coding of Audio-visual Object: Video, ISO/IECJTC/ SC29/ WG11, Jan. 1999.

[2] T. Sikora, “The MPEG-4 Video Standard Verification Model”, *IEEE Trans. Circuits and Systems for Video Technology*, **7**(6), 1997, 19-31.

[3] MPEG-4 Overviews, *http://www.chiariglione.org/mpeg/ standards/mpeg-4/mpeg-4.htm*

[4] N. Brady, F. Bossen, and N. Murphy, “Context-based Arithmetic Encoding of 2D Shape Sequences”, in Proc. of IEEE International Conference on Image Processing, **1**, 1997, 29-32.

[5] J. W. Stroming, Y. Kang, T. S. Hung, and S. M. Kang, “New Architectures for Modified MMR Shape Coding “ISCAS’97, **2**, 1997, 1205-8.

[6] M. J. Chen, Y. P. Hsieh, and Y. P. Wang, “Multi-resolution Shape Coding Algorithm for MPEG-4”, IEEE Trans. Consumer Electronics, **46**(3), 2000, 505-513.

[7] Y. T. Hwang, Y. C. Wang, and S. S. Wang, “An Efficient Shape Coding Scheme and its Codec Design”, *IEEE Workshop on Signal Processing Systems*, 2001, 225-232.

[8] Madisetti A. and Willson N., “A 100MHz 2-D 8×8 DCT/ IDCT processor for HDTV applications”, *IEEE Trans. Circuits Systems for Video Technology*, **5**(2), 1995, 158-164.