A NOVEL ALGORITHM OF SPATIAL SCALABILITY FOR SCRAMBLED VIDEO

YuanZhi Zou[1, 2], *Wen Gao*[1]
[1]Institute of Computing Technology, Chinese Academy of Sciences,
JDL, P.O. BOX 2704, Beijing, 100080, P.R.China
E-mail: {yzzou, wgao, tjhuang}@jdl.ac.cn
[2]Graduate School of the Chinese Academy of Sciences

ABSTRACT

Although many researches have investigated transparent scrambling video techniques, an issue of which transcoding relates to downsizing scrambled video without unscrambling has hardly been solved in single-layer MPEG-2 coded video content. In the paper, a novel algorithm based on approximation for compressed domain linear operation and using a mix of linear transformation of coefficient values in DCT for scrambling video is proposed for downsizing the scrambled video by a factor of two. Experimental results show that the proposed algorithm can reduce computational complexity and efficiently degrade video quality. Compared to the existing scrambling video methods, this algorithm not only can scramble video that is compliant to block-coded format but also can realize downsizing the scrambled video in compressed domain by a factor of two without unscrambling and impact on the performance of the compression.

1. INTRODUCTION

Content providers for high-quality mainstream applications, such as DTV and DVD, have already adopted single-layer MPEG-2 Video coding as the default format, hence a large number of MPEG-2 coded video content already exists. To access these existing MPEG-2 video contents from various devices with varying terminal and network capabilities, transcoding is needed. When content providers may provide pay-per-view services, the critical problem associated with the transcoding of scrambled bit streams include breaches in security by unscrambling and re-scrambling within the network, as well as computational issues. Although there are [1][3] [4] with a secure scalable streaming format that combines scalable coding techniques with a progressive encryption technique, handling this for non-scalable video and streams scramble with traditional encryption techniques is still an open issue. As pointed out in [2], for many real-world applications such as pay-per-view, although the content data rate is very high, the monetary value to the bits is low; therefore, "very expensive attacks are not interesting to adversaries" and "hence, light-weight encryption algorithms which can provide sufficient security level and have an acceptable computational cost are attractive to MPEG video applications". When scrambling video for non-scalable video with format-compliance, then processing it without unscrambling in the devices such as gateways, multipoint control units and servers, we must require an algorithm that can implement spatial scalability without unscrambling the scrambled video bit streams for access in the high or low definition devices. Scrambling video techniques include the following:

- *Format compliance* [1][2][3][4][5][6]. As can inherit network friendliness, error resilience as well as adaptation to protocols designed for the transport of standards-based compressed bit streams.
- *Selective encryption schemes*. Permutation of AC coefficients and encrypted DC coefficient [10] using DES, encryption method for MV field [8] using block ciphers such as AES and DES in the appropriate mode or by using stream ciphers and so on.
- *Spatial shuffling of codewords in compressed bit streams.* MB as basic shuffling unit, code 8×8 block as basic shuffling unit and run-level codeword as basic shuffling unit [3] [4] [5].

Although as above the methods of scrambling video can keep format-compliance, it cannot support spatial scalability without unscrambling because of destroying the positions or values of the critical information fields, such as AC and DC coefficients. When exploiting the current scrambling algorithms for video, we can find that lightweight security can realize downsizing the scrambled video in compressed domain by a factor of two without unscrambling with possibility for maintaining format-compliance to the syntax of video and for spatial scalability in no-scalable video and streams. In the paper, we provide a method that utilize approximation for compressed domain linear operation and a mix of linear transformation of coefficient values in DCT for scrambling video can solve the as above problem and implement it.

This paper is organized as follows. In Section 2, an algorithm including scrambling and downsizing is described. In Section 3, the experiment and result is shown. Finally, we summarize our major findings and outline our future work.

2. ALGORITHM

For downsizing scrambled video, there are two steps. One step is scrambling video of which format is convenient for the other step that is to downsize the scrambled video by a factor of two without unscrambling the scrambled video.

2.1 Scrambling video

[6] Pazarci, M., et al. proposes a method of a mix of linear transformation of pixel values containing a random mixture of brightness, contrast changes and negative/positive switching with different parameters for each of the RGB components. As is a point operation prior to the MPEG encoding, the scrambling

operation itself does not require any memory, and it can be implemented at high speed at low-cost. [7] Ci wang, et al. has applied the idea in the [6] to scrambling video in DCT domain of which coefficients in I frame and intra blocks in P and B can be scrambled, as may cause bit rate fluctuation because of encoding AC coefficients by variable length code, and create scrambled parameters overhead that can scramble DC coefficients and AC coefficients. Although the video bit streams scrambled by above the two methods can be decompressed, and then be downsized in the spatial domain, as needs descrambled and then scrambled, so the method may be insecure. As follows, a scrambling algorithm that can use a method of a mix of linear transformation of DC coefficients is proposed and can support downsizing in DCT by a factor of two without unscrambling.

For simplicity, we only exploit the downsizing by a factor of two in each dimension and the video sampling scheme is 4:2:0. When the DC coefficient is set zero in al l intra blocks, although video program display as shown in Figure 1 can be viewed, users must pay money for high quality picture. It is important to realize that overprotection of the content will not only increase cost but also make the product difficult to use, and may offend the user. Hence, lightweight scrambling algorithms that can provide sufficient security level and have an acceptable computational cost are attractive to MPEG video.

In terms of wide range for scrambling video, we select an MB as a processing unit. When video bit streams is scrambled, only intra block DC coefficient in C_y, C_r and C_b block must be scrambled. As exist two virtues, one can reducing computational complexity, the other can provide convenience for downsizing the scrambled video without unscrambling. For pth MB in a frame and 4:2:0 sample scheme, there are four luminance blocks such as (p+1)th, (p+2) th, (p+3)th , (p+4)th and one chromatic Cr blocks (p+5)th, one chromatic Cb blocks (p+6)th. We may change the DC coefficient value distribution according to some rules, and then control the extent of variation in coefficient value distribution for different degraded visual quality.

$$\chi_{pl}^{o}=\begin{cases}\alpha_p(1-\beta)-\beta\chi_{pl}^{i} & D=1\\ \alpha_p(1-\beta)+\beta\chi_{pl}^{i} & D=0\end{cases}\quad 0\le\beta\le 1$$

$$p\ni[1,M]\qquad l\ni[p+1,p+6] \tag{1}$$

Where α_k is impact factor in the pth MB in the frame, D is the operation-direction bit, M denotes the number of MB in a frame, χ_{pl}^{o}, χ_{pl}^{i} denotes the lth scrambled and unscrambled DC coefficient value in the pth MB.

$$\alpha_p=(\frac{k_1}{4}\sum_{n=1}^{4}\left|DC_{p,p+n}\right|+k_2\sum_{n=5}^{5}\left|DC_{p,p+n}\right|+k_1\sum_{n=6}^{6}\left|DC_{p,p+n}\right|)\times\frac{1}{3}$$

$$k_1+k_2+k_3=1,\ p=1,\ldots,M \tag{2}$$

Where $DC_{p,p+n}$ is the (p+n) th 8×8 *DCT in the* pth MB for a frame, n corresponds to video sampling scheme, $\left|\ \right|$ is the absolute value operation.

For every MB in a frame, D, α_p , and β references are overhead that are required、 encrypted and transferred for scrambling, When controlling three parameters as above, efficient video degraded quality can happen.

2.2 Downsizing video

To downsize the video scrambled by the algorithm in section 2.2, we use approximation for compressed domain linear operation.

For simplicity, we may study downsizing the video by a factor of two that means every four 8×8 DCT blocks give one output block. For two consecutive 8×8 DCT blocks that consist of U and V , downsizing by a factor of two can map U and V into a 8×8 DCT block O , O^j , U^j and V^j are jth 8-dimension column vectors in O , U and V , respectively. The implemented equation [8] is illustrated in (3), (4).

$$O^j=AU^j+BV^j \tag{3}$$

$$o_i^j=\sum_{t=1}^{8}(A_{it}\mu_t^j+B_{it}\nu_t^j) \tag{4}$$

In a 8×8 DCT, where O^j , U^j , V^j denotes the jth 8-dimension column vector, the tth coordinates are denoted by μ_t^j , and ν_t^j , $t=1,\ldots,8$, $j=1,\ldots,8$., o_i^j is the ith column and jth row coefficient. A and B 8×8 matrices [8] are given as (12) and (13). When downsizing the unscrambled video bit streams and the scrambled video bit streams, via (1), (3), (4), (12) and (13), we have results as follows.

If $i=1$ and $j=1$, then

$$so_1^1=\frac{(-1)^D\times\beta(\mu_1^1+\nu_1^1)}{2}+\frac{\alpha_p(1-\beta)}{2}$$

$$o_1^1=\frac{(\mu_1^1+\nu_1^1)}{2} \tag{5}$$

If $j\neq 1$ then

$$so_i^j=(A_{i1}\mu_1^j+B_{i1}\nu_1^j)+\sum_{t=2}^{8}(A_{it}\mu_t^j+B_{it}\nu_t^j) \tag{6}$$

When $i=1,\ldots,8$, $j=2,\ldots,8$

$$o_i^j = (A_{i1}\mu_1^j + B_{i1}\nu_1^j) + \sum_{t=2}^{8}(A_{it}\mu_t^j + B_{it}\nu_t^j) \quad (7)$$

Where so_i^j and o_i^j are denoted as the ith row and jth column coefficient of output that results from downsizing the two consecutive U and V of a 8×8 DCT block in an MB block, respectively by a factor of two when scrambling and unscrambling.

Via (6) and (7), we can find that if $j \neq 1$, then so_i^j and o_i^j are same, when we can get three references relating to scrambling in section 2.1 and restore o_1^1 via (5), then downsizing the scrambled video without unscrambling can be implemented.

Next, we may assume four 8×8 DCT luminance blocks M_{p+1}, M_{p+2}, M_{p+3}, and M_{p+4} in the MB. $M_{t,p+1}^j$, $M_{t,p+2}^j$, $M_{t,p+3}^j$ and $M_{t,p+4}^j$ denote the tth column and jth row in the (p+1)th, (p+2)th, (p+3)th and (p+4)th 8×8 DCT in the MB block, respectively. Via (3), (4), (6), (12) and (13), when three times computing, we have results that result from downsizing the consecutive four 8×8 DCT luminance blocks as below:

$$three\_so_1^1 = \frac{(-1)^D \times \beta(M_{1,p+1}^1 + M_{1,p+2}^1 + M_{1,p+3}^1 + M_{1,p+4}^1)}{4}$$

+

$$\frac{\alpha_P(1-\beta)}{2} \quad (8)$$

$$three\_o_1^1 = \frac{(M_{1,p+1}^1 + M_{1,p+2}^1 + M_{1,p+3}^1 + M_{1,p+4}^1)}{4} \quad (9)$$

$$three\_so_i^j = three\_o_i^j,\ i = 1,\ldots,8,\ j = 2,\ldots,8 \quad (10)$$

Via (8) and (9), we have equation (11).

$$three\_so_1^1 = (-1)^D \times \beta \times three\_o_1^1 + \frac{\alpha_p(1-\beta)}{2} \quad (11)$$

Where $three\_o_i^j$, $three\_so_i^j$ is the output of respectively downsizing the scrambled, unscrambled consecutive four 8×8 DCT luminance blocks.

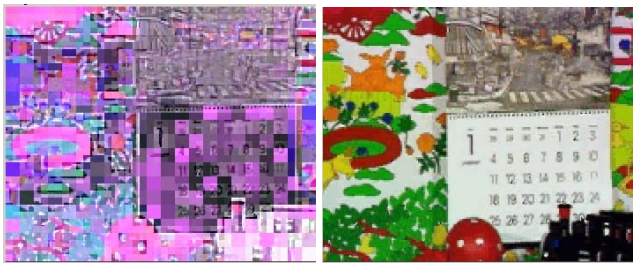
Similarly, we can process the four adjacent C_r and C_b blocks. Because of no-scrambling MV, the problems associated with mapping motion vectors can be addressed in the method of [9].

2.3 Unscrambling video

When a user has required the output of downsizing the scrambled video bit streams, at the user side, via (11), DC coefficients can be restored with scrambled parameters (D, α_p, and β) that can be encrypted by DES, as can cause to unscramble the downsizing of scrambled video.

3. EXPERIMENT

When a user has required the output of downsizing the scrambled video bit streams, at the user side, via (11), DC coefficients can be restored with scrambled parameters (D, α_p, and β). We implemented ours algorithm on the platform of MPEG-2 encoder and decoder. It is possible to change related scrambling parameter and efficiently degrade the video quality. Two sequences that are "mobile" and "template" with 352*288 and sampling scheme 4:2:0 are tested, and results are illustrated in Figure 1.



(a) Mobile Sequence

(b) Template Sequence

Figure 1. Example sequences: (from top to bottom) mobile and template sequence, (from right to left) after downsizing the scrambled video (352*288) by a factor of two, scrambled and unscrambled video display (176*144), respectively

For every GOP of 352*288 MPEG-2, there are at least 396 Intra MB and scrambling parameters include 1188 different parameters that may keep invariable. There exists difference for scrambling parameters between adjacent GOPs. Because computer vision technology is still naive at present, using the fierce-force-attack for one MB, there are at least $(1024)^6$ of the trials. In addition, there exists parameter difference between adjacent MBs. With the parameters scrambled key often being refreshed, so our algorithm can provide a certain extent of security level for preventing plaintext-attack and fierce-force-attack.

On account of transmitting the encrypted scrambling parameters that include D, α_p, and β, for 352*288 MPEG-2 video, there is 2% overhead. DC coefficients can be encoded by FLC (Fixed Length Code), so using a mix of linear transformation of DC coefficient values for scrambling video cannot cause bit rate

fluctuation and has no impact on the performance of the video compression.

$$A = \begin{vmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & -0.125 \\ -0.125 & 0.5 & 0.25 & 0 & 0 & 0 & -0.125 & -0.125 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & -0.25 & 0 \\ 0.125 & -0.125 & 0.25 & 0.25 & 0 & -0.125 & -0.125 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & -0.25 & 0 & 0 \\ -0.125 & 0.125 & -0.125 & 0.25 & 0 & -0.25 & 0.125 & 0 \end{vmatrix} \quad (12)$$

$$B = \begin{vmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0.125 \\ 0.125 & 0.5 & -0.25 & 0 & 0 & 0 & 0.125 & -0.125 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & -0.25 & 0 \\ -0.125 & -0.125 & -0.25 & 0.25 & 0 & -0.125 & 0.125 & 0 \\ 0 & 0 & 0 & -0.5 & 0 & 0.25 & 0 & 0 \\ 0.125 & 0.125 & 0.125 & 0.25 & 0 & -0.25 & -0.125 & 0 \end{vmatrix} \quad (13)$$

4. CONCLUSION

Our main contributions have two facets. One is to provide a new scrambling video algorithm; the other is to combine the approximation for compressed domain linear operation with the algorithm as above to solve the problem of downsizing the scrambled video by a factor of two without unscrambling in single-layer MPEG-2 coded video content and provide theoretical and experimental verification.

Our algorithm can provide a seamless access and lower computing overhead for high or low definition display devices, using a copy of video program by transcoding the scrambled video program, as can not be implemented by current other scrambled algorithms.

In future work, we will study how to further enhance the secure level of the scrambled video by the proposed algorithm.

5. REFERENCES

[1] Yu, H.H. "On scalable encryption for mobile consumer multimedia applications," Communications, 2004 IEEE International Conference on, Volume: 1, 20-24 June 2004

[2] Shi and B. Bhargava "A fast MPEG video encryption algorithm," in Proc. ACM Multimedia'98, 1998, pp. 81–88.

[3] MPEG4 IPMP FPDAM, ISO/IEC 14 496-1:2001/ AMD3, ISO/IEC JTC1/SC 29/WG11 N4701, March 2002.

[4] W. Zeng and S. Lei "Efficient frequency domain video scrambling for content access control," in Proc. ACM Multimedia'99, Orlando, FL, Nov. 1999, pp. 285–294.

[5] Chun Yuan, et al. Efficient and fully scalable encryption for MPEG-4 FGS, This paper appears in: Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on

[6] Pazarci, M., Dipcin, V. "A MPEG2-transparent scrambling technique," Consumer Electronics, IEEE Transactions on, Volume: 48, Issue: 2, May 2002 Pages: 345 – 355

[7] Ci Wang, et al. "A DCT-based MPEG-2 transparent scrambling algorithm," Consumer Electronics, IEEE Transactions on, Volume: 49, Issue: 4, Nov. 2003 Pages: 1208 – 1213

[8] Merhav, N., Vasudev, B. "A multiplication-free approximate algorithm for the inverse discrete cosine transforms," ICIP 99. Proceedings. Volume: 2, 24-28 Oct. 1999 Pages: 759 -763 vol.2

[9] N. Bjork and C. Christopoulos "Transponder architectures for video coding," IEEE Trans. Consumer Electron., vol. 44, pp. 88-98, Feb. 1998.

[10] Lei Tang "Methods for encrypting and decrypting MPEG video data efficiently," in Proceedings of the fourth ACM international conference on Multimedia (February 1997)