Drift-free Video Coding for Privacy Protected Video Scrambling

Roya Choupani
Faculty of Electrical Engineering,
Mathematics and Computer Science
Delft University of Technology
Delft, the Netherlands
e-mail: r.choupani@tudelft.nl

Stephan Wong
Faculty of Electrical Engineering,
Mathematics and Computer Science
Delft University of Technology
Delft, the Netherlands
e-mail: J.S.S.M.Wong@tudelft.nl

Mehmet Tolun
Department of Electrical Engineering
Aksaray University
Aksaray, Turkey
e-mail: mehmet.tolun@aksaray.edu.tr

***Abstract*—With video surveillance systems becoming ubiquitous nowadays, protecting people's privacy raises an increasingly serious concern. Video streaming with privacy protection requires modifying parts of the video content. This modification should provide the possibility of unprotected access to the video if the user is authenticated through a private key. However, any modification in the content of a video can result in a drift error and deteriorate the quality of the reconstructed video. In addition, it is required that the privacy protection does not adversely affect the computational complexity and the coding efficiency in terms of bitrate. In this work, we propose a drift-free method for scrambling the privacy protected regions of the frames while preserving the coding efficiency. Our proposed method provides the possibility of utilizing private keys for restricting unauthorized access to the private contents of the video with a small increase in the computational complexity of the encoder and decoder. The experimental results indicate that our proposed drift-free method can achieve a higher coding efficiency of 0.6 dB on average compared to similar methods.**

***Index Terms*—Privacy Protection in Video, Video Scrambling, Restricted Video Coding.**

I. Introduction

With the fast progress in multimedia technologies, video surveillance has obtained a widespread use. This excessive use of multimedia in general and video in particular, has given rise to many concerns about the privacy of individuals [1][2]. Among the solutions provided for the privacy protection in video is scrambling parts of the video frames corresponding to the private information such as the identity of the people in the scene [3][4][5][6]. The scrambling is performed by using a private key which can also be used for unscrambling the video. This means that the video should be decodable in both scrambled and unscrambled forms at the receiver side. Scrambling is performed by inverting the sign of AC coefficients [7], dividing the coefficients by a scrambling matrix generated at the server side using a private key [3], or by means of a seed number generating a random sequence which is used in performing XOR-operation on the coefficients [8]. The state-of-the-art video coding standards such as H.264, make use of Motion Compensated Temporal Filtering (MCTF) methods for eliminating temporal redundancy [9][10]. Hence, the inter-coded frame blocks are motion compensated before applying the Discrete Cosine Transform (DCT). If the area to be scrambled is used (even partially) as the reference for a block from a succeeding frame(s), the reconstructed video at the receiver will suffer from a frame quality degradation if the decoder does not have the necessary key to unscramble the reference frame. This degradation is accumulated in the subsequent frames and results in the drift error until an intra-coded frame (I frame) is reached. The drift error which is the result of mismatches between the reference frames at the encoder and the decoder, is one of the most important challenges in the privacy protection of video through scrambling. This paper addresses the drift error problem in privacy protected video coding and proposes a new solution for the problem. The paper is organized as follows: In Section II we review the related previous work and the state-of-the-art in privacy protection in videos. Section III introduces our proposed method, followed by the experimental results in Section IV. Finally in Section V we draw our conclusions and indicate possible directions of improvement.

II. Related Work

Apart from the methods which are based on the intra-coding of video frames without utilizing MCTF [11], the drift error caused by scrambling Region of Interest (ROI) areas is addressed in two different ways. The first group of methods is based on letting the drift error to happen either assuming that the scrambled frame range contains the entire GOP, or its effect is negligible.

In [12] the authors protect video content privacy by scrambling the entire frame by means of a context-aware middleware. The scrambling is carried out by by pseudo-randomly flipping the ac/dc coefficients of macro-blocks (MBs) in intra-coded frames only. Hence, when the key to unscramble the intra-coded frame is not available, the entire GOP becomes scrambled. In fact the reconstruction of the inter-coded frames of a scrambled GOP suffers from the drift error however, since these frames are not intended to be clearly decoded, the drift error becomes irrelevant here. Dufaux et al. [7] change the AC coefficients using a pseudo-random sequence generated by a seed number. They toggle the coefficients as:

$$qAC_{coef} = \begin{cases} -qAC_{coef} & \text{if RandomBit=1} \\ +qAC_{coef} & \text{otherwise} \end{cases}$$

without considering the effect of the drift error.

The second group of methods are based on avoiding the drift error. In [4] the authors propose a restricted video coding scheme to avoid the drift error. The main idea in their proposed method is restricting the search area during motion estimation not to include the scrambled areas. In this way the scrambled areas are not used as reference for any MB and hence, the drift error is avoided when the user is not authorized to see the protected areas. They propose three different methods, Mode Restricted Intra Prediction (MRIP), Search Window Restricted Motion Estimation (SWRME), and Boundary Strength Restricted Deblocking Filtering (BSRDF), to handle the intra-prediction in I frames, and inter-frame prediction modes to avoid using scrambled areas for motion compensation. Tong et al. [6] improve the MRIP method proposed by Dai et al. [4] assuming that the most probable blocks to be utilized in intra-prediction mode are left and top blocks of the current block. Hence, if the left or top block of the current block is in the scrambled area, they forbid using intra 4×4 prediction mode for it. Wang et al. [13] use a similar method to scramble the video however, they reduce the bitrate overhead by choosing the prediction modes of the 4×4 blocks around the boundary of the privacy area instead of forbidding intra-prediction. In [14] a bit mask is proposed to indicate the location of blocks from the privacy area. Our proposed method is a drift-free scheme however, the encoder does not need to restrict the search areas during motion estimation. In this way, the complexity of the encoder is reduced while the coding efficiency is slightly improved. To justify this improvement, we can consider the case when the most similar area to a block partly overlaps with a scrambled area. However, if overlapping is not allowed during motion estimation, the residual values after motion compensation can be larger.

III. Proposed Method

Our proposed method utilizes the linearity characteristic of the cosine transform. The main issue addressed in the proposed method is eliminating the drift error in protecting the privacy of the people appearing in the video, while preserving the video coding efficiency. The methods proposed in the literature perform motion compensation in the original video, and then scramble the quantized and DCT-transformed residues. We propose performing motion compensation using scrambled frames. The block diagram of our encoder is depicted in Figure 1. The scrambling is carried out by post-multiplication of the 8×8 matrix of quantized coefficients by matrix ***S*** which is defined in Equation 1.

$$S = sign\begin{pmatrix} -||key||^8 + 0.5 & 0 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & -||key||^1 + 0.5 \end{pmatrix} \quad (1)$$

where ***key*** is a positive integer of eight digits representing the secret key being used for scrambling and unscrambling the privacy protected parts of the frames, and $||.||^r$ refers to normalizing the r^{th} digit of an integer number to a decimal number between 0 and 1 as shown in Equation 2.

$$\begin{array}{ll} For & r = 1, .., 8 \\ & y = floor((x \pmod{10^r}) \times 10) \\ & ||x|| = \frac{y}{10} \end{array} \quad (2)$$

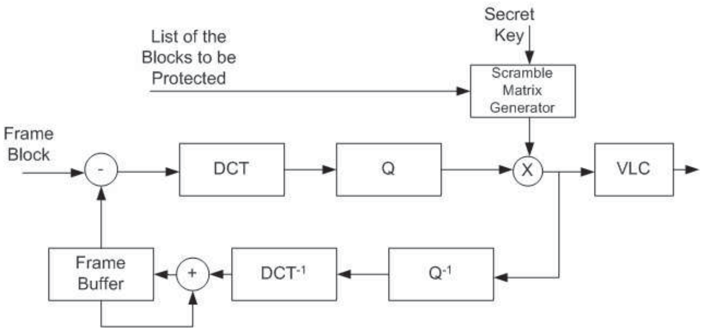

Figure 1. Frame coding block-diagram of the proposed method.

In fact the scrambling matrix is used for toggling the signs of the quantized DCT coefficients at the main diagonal position. The most significant digit of the ***key*** value toggles the sign of the DC value in the DCT coefficients. Changing the DC value scrambles the area completely however, the visual quality of the image degrades significantly. If a simple blurring of the protected area is sufficient, this digit should be less than 5. Given a block of motion-compensated coefficients, ***B***, the encoding process is as given in Equation 3.

$$EB = VLC(S \times Q(DCT(B))) \quad (3)$$

where *EB* is the encoded block, and Q and DCT refer to the quantization and the discrete cosine transform, respectively. Our assumption is that the locations of the blocks belonging to the privacy protected area are provided as shown in Figure 1. The decoding process can be carried out in two different cases as below:

- The user does not provide the decoding key. In this case the normal steps of decoding blocks are followed. Since the scrambled frames have been used as (part of) the reference areas, in the decoded frames the privacy protected areas appear as scrambled.
- The user provides the decoding key which is used for creating matrix S. This matrix is used for unscrambling the protected areas as shown in Equation 4.

$$DB = Inv(DCT^{-1}(S)) \times DCT^{-1}(Q^{-1}(VLC^{-1}(EB))) \quad (4)$$

where DCT^{-1}, Q^{-1}, and VLC^{-1} are inverse DCT, inverse quantization, and inverse variable length coding, respectively. *DB* refer to the decoded blocks, and *Inv* is used to show the inverse of a matrix. An analytical proof of Equation 4 where we assume quantization is an invertible operation, is given in Equation 5. Replacing *EB* with its definition from Equation 3, we have:

$$
\begin{aligned}
DB =& Inv(DCT^{-1}(S)) \times DCT^{-1}(Q^{-1}(VLC^{-1}(VLC(\\
& S \times Q(DCT(EB)))))) \\
DB =& Inv(DCT^{-1}(S)) \times DCT^{-1}(Q^{-1}(S \times Q(DCT(EB)))) \\
DB =& Inv(DCT^{-1}(S)) \times DCT^{-1}(S \times DCT(EB)) \\
DB =& Inv(DCT^{-1}(S)) \times DCT^{-1}(S) \times EB) \qquad (5) \\
DB =& EB
\end{aligned}
$$

Despite the methods proposed in the literature which provide privacy protection through direct manipulation of the quantized coefficients, our proposed method uses matrix multiplication for modifying the coefficients. This lets us utilize the linear property of the DCT to design the decoder so that the unscrambling is done as the last step and after applying inverse DCT to the coefficients. Hence, the proposed method has a minimal impact on the encoding and decoding procedures. For instance, the proposed method does not need to consider if a non-protected block is motion-compensated (partly) using a privacy protected block or not. The decoding process is depicted in Figure 2. As shown in Figure 2, in both cases

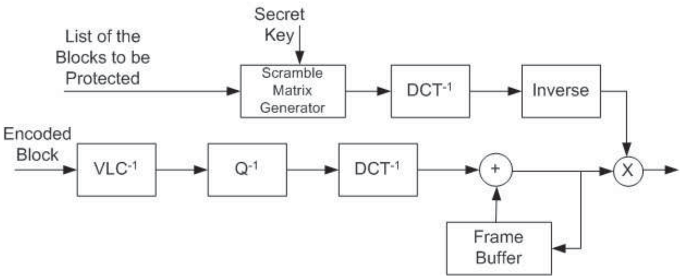

Figure 2. Frame coding block-diagram of the proposed method.

whether the decoding key is provided or not, the reference frames are unscrambled frames. Considering that the motion compensation in frame encoding is done using scrambled frames, no drift error is created by the proposed method. Moreover, there is no need to restrict the search area of during motion estimation process hence, the proposed method adds only minor modifications to the standard video coding methods. This means that the proposed method can be used with all state-of-the-are video coding standards.

IV. Experimental Results

In our proposed method we claim that the method does not suffer from the drift error yet its encoding performance in terms of rate distortion ratio is better than the state-of-the-art methods. Moreover, the encoding latency, and the coding complexity of the proposed method are less than the similar methods in literature. As benchmarks we are considering the method proposed by Tong et al [6].

A. Time Complexity Analysis

The proposed method uses a matrix multiplication operation to scramble the privacy related areas of the frames. In this subsection we analyze the processing cost of the scrambling operation with respect to the total frame encoding time. Since we assume only some parts of each frame is scrambled, we consider the parameter ρ as the probability of a block falling into the private area of the frame. Without considering computationally less intensive steps such as interpolation filtering, deblocking, and in-loop filtering, the encoding time of a block is given as shown in Equation 6.

$$T_{Encode} = T_{ME} + T_{DCT} + T_Q + \rho \times T_{MM} + T_{VLC} \qquad (6)$$

where T_{ME} is the time spent for motion estimation, T_{DCT} is the time for applying the DCT, T_Q is the time for quantization, T_{MM} is the matrix multiplication time, and T_{VLC} is the time for variable length coding. The extra step added by the proposed method increases the time complexity of the encoder by $O(n^{2.373})$ [15] where *n* is the size of the matrix. However, the scrambling matrix in the proposed method is a diagonal matrix consisting of ± 1. This property reduces the matrix multiplication to eight sign changes. Moreover, considering the following scenarios for the privacy protected parts of a frame, we estimate $\rho = 0.1$.

- A security assistant/officer may want to have his/her face image to be hidden from un-authorized viewers.
- A driver may ask for the protection of his/her face from in videos.
- The license plates of the cars passing by an accident area should be hidden in a public video stream.

The DCT is performed as a matrix-vector product [16] as $y = Tx$. The fast transform is carried out by the factorization of T into a product of sparse structured matrices. Vashkevich et al. [17] presented a fast DCT algorithm which uses 32 multiplication and 81 addition for a 16-point data. The time needed for the DCT in comparison to the matrix multiplication used by the proposed method, considering that the motion estimation accounts for about 60% of the video coding time, and the fact that the scrambling is applied to a small portion of the frame blocks, the impact of the proposed method on the total video coding time is minor.

In decoding stage, the inverse matrix computation ($inv(DCT^{-1}(S))$) is performed only once as long as the key value remains the same. Therefore, without considering the motion estimation, the time complexity values of the encoder is valid for decoder too.

B. Bit-rate Overhead Analysis

In order to avoid the drift error we propose motion compensating the blocks without forbidding the partially scrambled area of the reference frames as explained in Section III. Although in general there will be a slight reduction in the coding efficiency compared to the case when video is encoded without scrambling, we claim that the proposed method provides better bit overhead saving than MRIP method proposed in [6]. This improvement is explained by considering the fact that the MRIP method restricts the search area while the proposed method searches everywhere including the areas searched by MRIP. In our first set of experiments we have compared the performance of the proposed method with MPEG encoder without scrambling the frames. The results depicted in Figures 3 and 4 show the performance of our proposed method in two video sequences. Our experimental results in all test

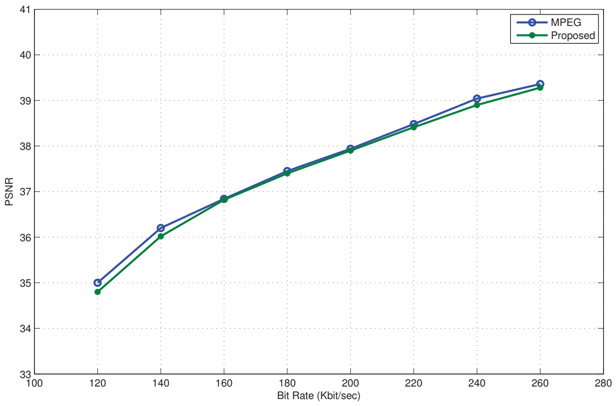

Figure 3. Rate-Distortion comparison of the proposed method and MPEG4 encoder using Foreman video sequence.

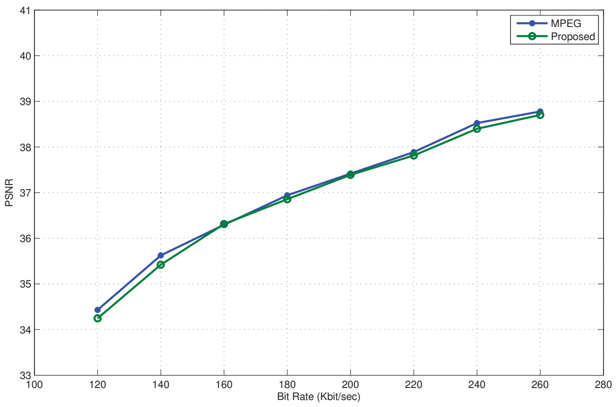

Figure 4. Rate-Distortion comparison of the proposed method and MPEG4 encoder using Container video sequence.

video sequences indicate that the performance of the proposed method is comparable with and only slightly less than MPEG4 video coding standard where scrambling is not applied.

In our second set of experiments we have compared the performance of our method with the case when the search area is restricted to non-scrambled areas. Figures 5 and 6 depict the rate distortion comparisons of the proposed method and the method proposed in [6] using '*Foreman*' and '*Container*' sequences, respectively. Our experiments with sample video sequences reveal that some of the blocks are motion compensated with areas which partially overlap with scrambled areas. This observation explains the improvement in coding efficiency of the proposed method which amount to 0.6 dB on average. Figures 7 and 8 depict the result of scrambling the privacy area and corresponding motion vectors, respectively. It is important to note that despite scrambling the face area, motion vectors do not show significant increase in size which justifies the

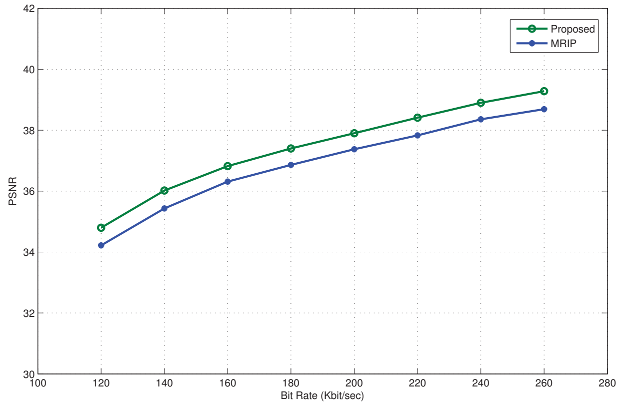

Figure 5. Rate-Distortion comparison of the proposed method and MRIP method using Foreman video sequence.

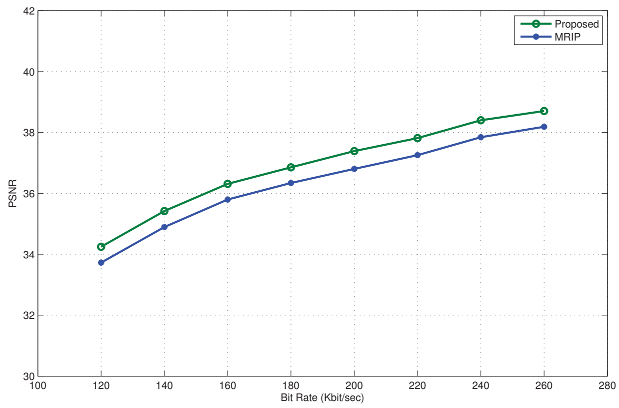

Figure 6. Rate-Distortion comparison of the proposed method and MRIP method using Container video sequence.

performance improvement of the proposed method. Our drift-free method has advantages over methods which include the whole frames in an entire GOP to avoid quality degradation due to the drift error [12]. In this case, the obvious advantage is that only privacy areas are scrambled in the relevant frames in our proposed method whereas, in these methods the entire frame, and some of the irrelevant frames (frames with no privacy importance) are also scrambled.

It should be noted that in our design, the decoder needs to know the location of privacy area or ROI. We assume this information is transmitted as a binary map which includes one bit for each macro-block. This corresponds to 396 bits per frame in CIF format. Since the main interest of this work is eliminating drift error due to scrambling, we have not considered the communication of the so-called binary map.

V. Conclusions

A new drift-free method for scrambling videos for privacy is proposed. The proposed method addresses the quality degradation due to the use of scrambled areas of a frame in motion compensating blocks of succeeding frames. Despite the methods proposed in the literature, our proposed method

Figure 7. Sample Frame from Foreman Video (left), with Scrambled Private Area (right).

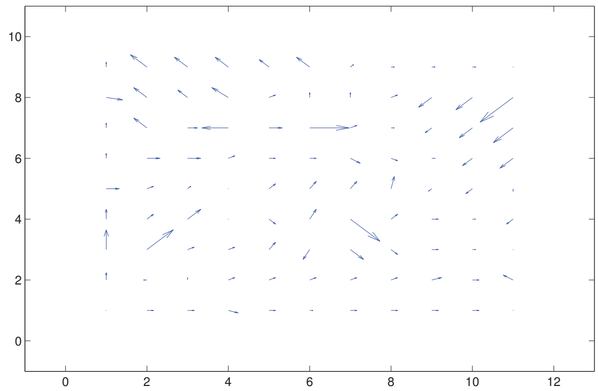

Figure 8. Motion Vectors of the Second Frame from Foreman Video Scrambled using the Proposed Method.

does not forbid utilization of the scrambled areas as reference areas. Besides, our method is capable of reconstructing the video both when the private key is available and not without any degradation in the frame quality. Our experimental results indicate that the performance of our method is comparable to standard video coding methods with only a very slight decrease in rate distortion ratio.

An important feature of the proposed method is that the scrambling and unscrambling steps are independent from the motion estimation/motion compensation and transformation of the residues. Therefore any encoding in the form of a matrix multiplication can be utilized. However, since the scrambling matrix consists of positive and negative ones, the performance degradation is limited. Moreover, the method is compatible with all video coding standards as it is applied after quantization step.

REFERENCES

[1] A. Becker, A. Arnab, and M. Serra. Assessing privacy criteria for DRM using EU privacy legislation. *Proceedings of the 8th ACM workshop on Digital Rights Management*, pages 77–86, 2008.

[2] J.R. Troncoso-Pastoriza, P. Comesaa, L. Prez-Freire, and F. Prez-Gonzlez. Videosurveillance and privacy: Covering the two sides of the mirror with DRM. *Proceedings of the Nineth ACM Workshop on Digital Rights Management*, pages 83–94, 2009.

[3] Y.H. Sohn, H.C. Huang, and S.Y. Wang. Video scrambling and fingerprinting for digital right protection. *International Symposium on Computer, Consumer and Control*, pages 471–474, 2012.

[4] L. Tong, F. Dai, Y. Zhang, and J. Li. Restricted H.264/AVC video coding for privacy protected video scrambling. *Journal of Visual Communications and Image Representation*, pages 479–490, 2011.

[5] H. Sohn, W. D. Neve, and Y. M. Ro. Privacy protection in video surveillance systems: Analysis of subband-adaptive scrambling in JPEG XR. *IEEE Transaction on Circuits and Systems for Video Technology*, 21(2):170–177, 2011.

[6] L. Tong, F. Dai, Y. Zhang, and J. Li. Restricted H.264/AVC video coding for privacy region scrambling. *Proceedings of 17th IEEE International Conference on Image Processing*, pages 2089–2092, 2010.

[7] F. Dufaux and T. Ebrahimi. H.264/AVC video scrambling for privacy protection. *IEEE International Conference on Image Processing*, pages 1688–1691, 2008.

[8] Y. Kim, S. H. Jin, T. M. Bae, and Y. M. Ro. A selective video encryption for the region of interest in scalable video coding. *IEEE Region 10 Conference (TENCON 2007)*, pages 1–4, 2007.

[9] H. Schwarz, D. Marpe, and T. Wiegand. Analysis of hierarchical b-pictures and mctf. *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1929–1932, 2006.

[10] ISO=IEC 144962. *Coding of audio-visual objects*. 1998.

[11] H. Sohn, W. D. Neve, and Y. M. Ro. Region-of-interest scrambling for scalable surveillance video using JPEG XR. *Proceedings of the 17th ACM international conference on Multimedia (MM 09)*, pages 861–864, 2009.

[12] J. Shen, Y.-J. Cai, and L. Luo. A context-aware mobile web middleware for service of surveillance video with privacy. *Multimedia Tools and Applications*, pages DOI 10.1007/s11042–014–2036–9, 2014.

[13] F. Kurugollu Y. Wang, M. ONeill. Privacy region protection for h.264/avc by encrypting the intra prediction modes without drift error in i frames. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2964–2968, 2013.

[14] F. Kurugollu Y. Wang, M. ONeill. Adaptive binary mask for privacy region protection. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1127–1130, 2012.

[15] V. V. Williams. Multiplying matrices in $O(n^{2.373})$ time. *Stanford University*, 2014.

[16] R. J. Cintra and V. S. Dimitrov. The arithmetic cosine transform: Exact and approximate algorithms. *IEEE Transaction on Signal Processing*, 58(6):3076–3085, 2010.

[17] M. Vashkevich and A. Petrovsky. A low multiplicative complexity fast recursive DCT-2 algorithm. *CoRR*, abs/1203.3442, 2012.