Fast Block Direction Prediction for Directional Transforms

Gabriel Beltrão, Julio León, *Member, IEEE,* Rangel Arthur, *Member, IEEE,* and Yuzo Iano, *Senior Member, IEEE*

***Abstract*—DCT-based transforms are widely adopted for video compression. Recently, many authors have highlighted that prediction residuals usually have directional structures that cannot be efficiently represented by conventional Discrete Cosine Transform (DCT). Although directional transforms have superior performance over the conventional DCT, for application in video compression it is necessary to evaluate increase in coding time and complexity for its implementation.**

This paper proposes a fast algorithm for estimating blocks directions before applying directional transforms. The encoder identifies predominant directions in each block, and only applies the transform referent to that direction.

The algorithm can be used in conjunction with any proposed algorithm for directional transforms that uses the RDO process for selection of the direction to be explored, reducing implementation complexity to similar levels when only conventional DCT is used. For the tested proposal, estimated reduction in total coding time would be 37.5% (4×4 blocks with 8 possible directions) and 47.4% (8×8 blocks with 16 possible directions).

***Index Terms*—HEVC, directional transform, video coding.**

I. Introduction

DCT-based transforms are widely adopted for video compression. Recently, many authors have highlighted that prediction residuals usually have directional structures that cannot be efficiently represented by conventional Discrete Cosine Transform (DCT).

In this scenario, several directional transforms have been proposed for use in video coding. Some of them use lifting-based techniques to change conventional transforms into directional ones. Another approach is to use pixels information to build the directional transform, resulting in data-dependent transforms. The last category is based on the reorganization of pixels accordingly to selected direction before applying the conventional transform. Most of them takes into account only residuals of spatial prediction (intra-frames). Even in recent documents published about the High Efficiency Video Coding (HEVC) [1], which is being developed as the successor of H.264/AVC [2], directional transforms are only applied on intra-frames residuals. In general, few studies have addressed directional transforms on temporal (inter-frames) residuals, which are also of fundamental importance in video compression. A good overview of directional transforms can be found in [3].

G. Beltrão, J. León and Y. Iano are with the Department of Communications, Faculty of Electrical and Computer Engineering, University of Campinas, Campinas, CEP 13083-852 São Paulo, Brazil. E-mail: {gabrielb, j.leon, yuzo}@decom.fee.unicamp.br

R. Arthur is with the Faculty of Technology, University of Campinas, Limeira, CEP 13484-332 São Paulo, Brazil. E-mail: rangel@ft.unicamp.br

Manuscript received August 20, 2012.

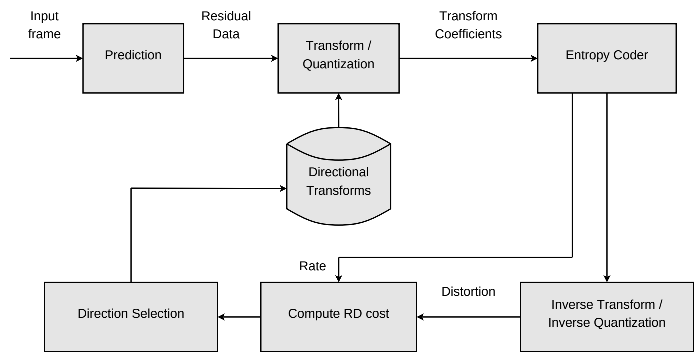


Figure 1: Rate-distortion optimized selection.

Although directional transforms have shown superior performance over conventional DCT, for application in video compression it is necessary to evaluate the increase in coding time and complexity for its implementation. Most of the recently proposed directional transforms [4–7] are based on the Rate Distortion Optimization (RDO) [8] process to select the direction/mode of the transform to be used for each block, as can be seen in Figure 1. This is a brute force technique in which, for each available option, the block is transformed, quantized, entropy coded, inverse quantized and inverse transformed. After that, rate-distortion cost is calculated and the option with the lowest cost is chosen. This process increases substantially encoder complexity and may be prohibitive for real-time applications.

In order to reduce this complexity, avoiding RDO calculations, a fast algorithm for estimating blocks directions is proposed. Before applying directional transform, the encoder identifies predominant directions in each block, and only applies the transform referent to that direction, or, in the case that there is no predominant direction, conventional DCT can be used. This method can be applied with any proposed algorithm for directional transforms that uses RDO process for selection of the direction to be explored, reducing implementation complexity to similar levels when only conventional DCT is used. This simplification is even more important when applied to HEVC, since HEVC coding structure is much more complex than H.264/AVC, as will be seen in the next section.

This paper is divided in five sections. Section II gives a brief overview of HEVC, based on tools already included in HM6 [9] (February 2012). Section III presents the algorithm for prediction of block directions while Section IV discuss its integration with directional transforms. In Section V, experimental results of software simulations are shown and, finally,

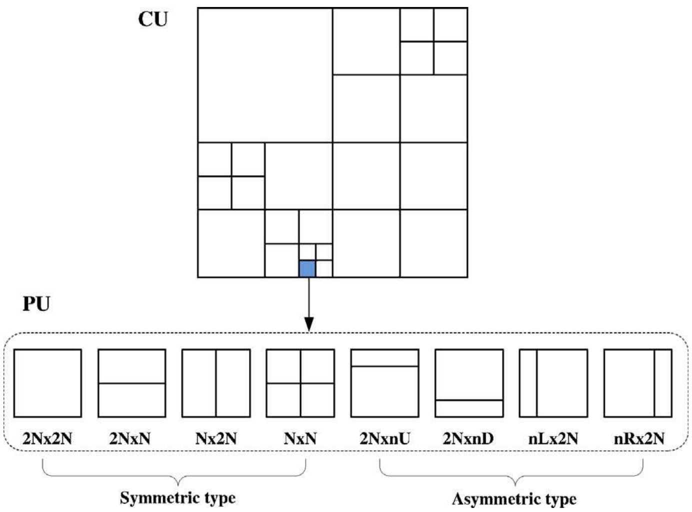


Figure 2: Division of CUs into PUs [22].

Section VI presents the conclusions about obtained results.

II. Overview of HEVC

The JCT-VC (Joint Collaborative Team on Video Coding) was created in January 2010, with the responsibility to receive and evaluate proposals, in a new standardization initiative known as HEVC (High Efficient Video Coding). It aims a compression gain at about 100% (half rate) compared with H.264/AVC, while maintaining the same visual quality.

In response to this call, 27 proposals were submitted and evaluated [10], showing significant improvements over H.264/AVC. A Test Model under Consideration (TMuC) was created [11], combining key elements of the better ranked proposals [12–17]. The TMuC was the basis for the first Test Model [18] and its software implementation [19], which has been improved at each JCT-VC meeting. Documentation for all meetings is available at [1]. A summary of the Call for Proposals and its results can be found in [20] [21].

The proposed structure for HEVC follows the traditional hybrid coding format, with spatial and temporal prediction, motion estimation and compensation, in-loop filtering, spatial transform of residuals, and adaptive entropy coding. Two encoder configurations (similar to H.264/AVC profiles) were defined: Low Complexity, with reasonably high compression performance while keeping codec complexity low; and High Efficiency (HE10), to obtain high compression performance [9].

One of the key elements for the superior performance of HEVC was the introduction of larger block structures with a flexible and hierarchical partitioning scheme called Treeblock Partitioning, originated from [16] [17] [22]. The Coding Unit (CU) is the basic unit of processing and is formed by non-overlapped square blocks of 8×8 up to 64×64 luma samples. Each CU can be recursively subdivided into four others CUs and so on until the minimum size of 8×8. Prediction methods (skip, inter or intra) are specified for the last depth CUs, which are now sub-divided into Prediction Units (PUs), according to the chosen method. PU is the basic unit of the prediction process and carry all information related to this process. Asymmetric shapes are now supported allowing better match with boundaries of real objects in the picture. Figure 2 shows the division of CUs into PUs. CUs are also divided into Transform Units (TUs), which are the basic unit for spatial transforms and quantization. TU sizes of 4×4 up to 32×32 luma samples are supported. Asymmetric shapes are also possible for inter-frames, according to PU partitioning mode. The possibility of arbitrarily choose units sizes allows the codec to be optimized for different content, applications and devices. Large blocks can be used to code large homogeneous areas better, using spatial redundancy. Smaller blocks can be used for very detailed regions or for low resolution devices.

Table I: Tools already included in HM6 [9]

Main	High efficiency 10 (HE10)
Coding units, Prediction units, and Transform units:	
CU quadtree structure (square block sizes 2Nx2N, for N=4, 8, 16, 32; i.e., up to 64x64 luma samples in size)	
Prediction units (for CU size 2Nx2N: for Inter, 2Nx2N, 2NxN, Nx2N; for Intra, only 2Nx2N and, for N=4, also NxN)	Prediction units (for CU size 2Nx2N: for Inter, 2Nx2N, 2NxN, Nx2N, and, for N>4, also 2Nx(N/2+3N/2) & (N/2+3N/2)x2N; for Intra, only 2Nx2N and, for N=4, also NxN)
TU tree structure within coding unit (maximum of 3 levels)	
Transform block size of 4x4 to 32x32 (always square)	Transform block size of 4x4 to 32x32 (always square for Intra; also non-square 4x16, 16x4, 8x32, 32x8 for Inter)
Spatial Signal Transformation and PCM Representation:	
DCT-like integer block transform; for Intra also a DST-based integer block transform (selected based on the intra prediction mode)	
Transforms can cross prediction unit boundaries for Inter; not for Intra	
Intra-picture Prediction:	
Angular intra prediction (35 directions)	
Planar intra prediction	
Inter-picture Prediction:	
Luma motion compensation interpolation: 1/4 sample precision, 8x8 separable with 6 bit tap values	
Chroma motion compensation interpolation: 1/8 sample precision, 4x4 separable with 6 bit tap values	
Advanced motion vector prediction with motion vector "competition" and "merging"	
Entropy Coding:	
Context adaptive binary arithmetic entropy coding	

Up to 35 (33 directional, besides Planar and DC) intra-frame modes are supported. An Advanced Motion Vector Prediction (AMVP) method is introduced by [16] which is adapted into the new coding structure. It allows the selection of the best predictor from a given set composed of three spatial motion vectors, a median motion vector and a temporal motion vector. DCT-like integer block transforms, associated with DST-based transform [23] (according to intra prediction mode), are specified for prediction residuals. Secondary transforms such as the Rotational Transform [16] [24], and Mode-Dependent

Secondary Transforms [25] are still under study. Increase in complexity is compensated by using Chen's fast algorithm for the DCT [26]. A similar CABAC entropy code scheme as in H.264/AVC is specified.

Table I summarizes some tools already included in HM6.

III. Block Direction Prediction

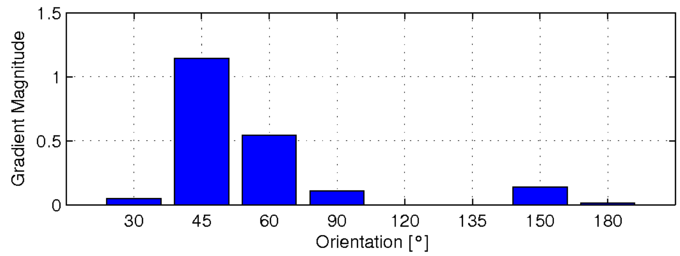
The algorithm was originally proposed in [27], and consists in finding predominant block direction from existing directions in each sample within the block. For each image sample $p(x, y)$, the gradient magnitude $m(x, y)$, and orientation $\theta(x, y)$ are pre-computed using pixel differences:

$$Gx_{x,y} = p(x+1, y) - p(x-1, y) \quad (1)$$
$$Gy_{x,y} = p(x, y+1) - p(x, y-1) \quad (2)$$
$$m(x, y) = \sqrt{Gx_{x,y}^2 + Gy_{x,y}^2} \quad (3)$$
$$\theta(x, y) = \arctan\left(\frac{Gy_{x,y}}{Gx_{x,y}}\right) \quad (4)$$

For optimization, the rooted sum of squares in (3) can be replaced by $|Gy_{x,y}| + |Gy_{x,y}|$, and the $\arctan$ function can also be replaced by $Gy_{x,y}/Gx_{x,y}$.

Then an orientation histogram is created, adding each sample, weighted by its gradient magnitude, to the correspondent direction bin. Peaks in the orientation histogram shows dominant directions of local gradients. A simple threshold can be used to select only one direction. Histogram discretization may be chosen according to the available set of directional transforms. Each possible direction of transform should be associated with an histogram bin.

(a) Residual block



(b) Orientation histogram

Figure 3: Residual block (a) and associated orientation histogram (b).

Figure 3 shows a residual block (a) and the associated histogram (b) for this block. It indicates the presence of an strong structure in the 45 degrees direction. The orientation histogram has 8 bins covering 180 degrees.

If the prediction algorithm takes a wrong decision about predominant direction within the block, the directional transform used will not be consistent with residual structure, resulting in the generation of unnecessary coefficients and degrading coding performance. Thus, thresholds must be carefully chosen so that direction prediction is made with a high degree of certainty. If there is no such certainty, it is prudent to use the conventional 2D-DCT.

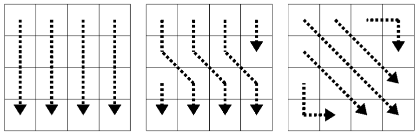
A similar algorithm, based on the computation of the orientation histogram, was already used to speed-up the selection of the intra-prediction modes in H.264/AVC [28], and recently in the HEVC [29].

IV. Integration with Directional Transforms

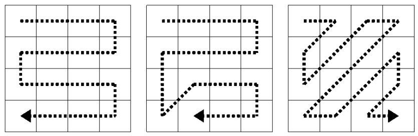
The algorithm was integrated with the directional transforms proposed in [30]. Kamisli and Lim analysed residuals generated by motion estimation and compensation process, and showed that this kind of data have different spatial characteristics from intensity images. These inter-prediction residuals have no regular or smooth regions as original images because prediction works effectively in these regions. Most energy is concentrated in regions difficult to predict (moving objects boundaries or edges), i.e. pixels of greater energy are concentrated along these edges, generating unidimensional structures with some predominant direction. This indicates that using 2-D transforms, with basis functions that have 2-D support, is not the best choice for such regions. The author proposes to use 1-D directional transforms with basis functions whose support the 1-D directional structures of motion compensation residuals.

Figure 4 shows proposed transforms and associated scan patterns [30]. They used 16 directions for 8×8 blocks (a) and 8 directions for 4×4 blocks (b), which together cover $180°$ (only some directions are shown; remaining directions are symmetric versions). Pixels are rearranged into vectors according to defined directions (arrows in the figure), and then 1-D DCT is applied in each one of these vectors. After the transform, coefficients are rearranged according the original position. The block is then scanned with the associated directional pattern so that significant coefficients (non-zero) are placed at the first positions of the vector that goes into the entropy coder.

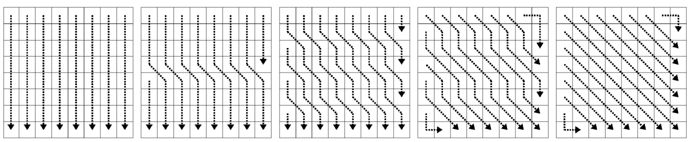
Integration was done so that always when the algorithm indicate some predominant direction, the performance of that directional transform should be better then all the others transforms, including conventional 2-D DCT. For analysis purposes, eight of the sixteen directional transforms were used: $30°, 45°, 60°, 90°, 120°, 135°, 150°$ and $180°$. Detected directions in the third and fourth quadrants (negative directions) were mapped to the correspondent positive ones (first and second quadrants). Actual calculated values are approximated to direction bins values according to some approximation criterion. Experimental results have shown that, for these eight directions, five degrees is a good criterion. For example, directions between $25°$ and $35°$ are counted in the $30°$ bin. Values outside the range of each bin are not used in the algorithm. If there is no strong predominance of one direction



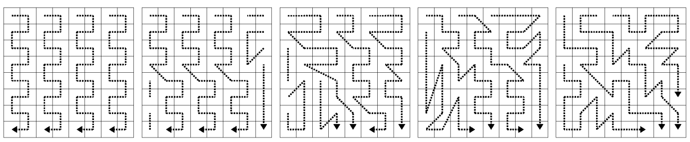
(a) Directional 1-D Transforms (4 × 4)



(b) Directional Scan Patterns (4 × 4)



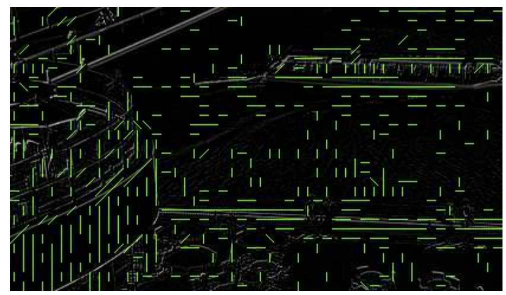
(c) Directional 1-D Transforms (8 × 8)



(d) Directional Scan Patterns (8 × 8)

Figure 4: Directional transforms and scan patterns [30].

(a) Original frame



(b) Detected directions

Figure 5: Original frame (a) and overplot (b) of detected directions (green lines).

over the others, the algorithm chooses to not indicate any direction, and conventional DCT can be used. As suggested in [28] and [29], the algorithm can also be used to select a small number of directional transforms as the candidates to be used in RDO calculations. This second approach can be very interesting in high complexity configurations of HEVC.

V. Experimental Results and Discussion

A. Block Direction Prediction

For an initial validation, the block direction prediction algorithm was implemented in MATLAB and tested with suggested sequences from JCT-VC for HEVC [31]. Residuals frames were generated and processed block by block (fixed size). Figure 5 and Figure 6 shows results obtained with two of the tested sequences. For this example, only four directions were used: $45°, 90°, 135°$ e $180°$. Green lines were plotted to illustrate identified directions.

B. Integration with Directional Transforms

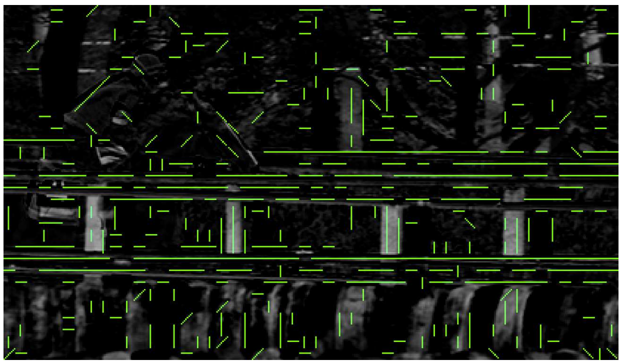
Figure 7 and 8 show residual blocks (a) with some directional structures and associated orientation histograms (b). Predominant direction is identified in the orientation histogram and indicated directional transform is used and compared with conventional 2-D DCT. It can be seen in Figure 7 and 8 (d) and (e), that energy is much more concentrated in transformed coefficients of the 1-D directional DCT. Figure 7 (e) and 8 (e) show retained energy in transformed coefficients after scan, for both 2-D DCT and the 1-D directional transform indicated by the direction prediction algorithm. When the directional 1-D DCT indicated is used, almost 90% of residual energy is concentrated in the first ten scanned coefficients, while, for the 2-D DCT, energy is spread over many more coefficients, which will considerably deteriorate coding performance. For example, in Figure 7 (e), the second coefficient of the 1-D directional transform retains more than 80% of all energy within the block, while, for the conventional 2-D DCT, 28 coefficients are necessary to get the same amount.

C. Complexity Analysis

Since the selection of the directional transform to be used is normally made by the RDO technique, in which all possible coding options are tested (in each test, residual blocks goes through spatial transform, quantization, entropy coding, inverse quantization, inverse transform and cost computation) and the one with the lower cost is chosen, the increase in coding time is assigned to this process and can be estimated by multiplying required time for test an option (direction) for the number of available options (directions).

In Kamisli experiments with H.264/AVC [30], for example, when only the conventional transform is used, RDO computations take about 6% (8×8 blocks) and 8% (4×4 blocks). Thus, estimated factor of increase in total coding time, ignoring the complexity of each transform itself, is 1.9 (16 additional transforms), and 1.6 (8 additional transforms). Decoding time does not increase since decoder only performs the directional transform signalled by the encoder.

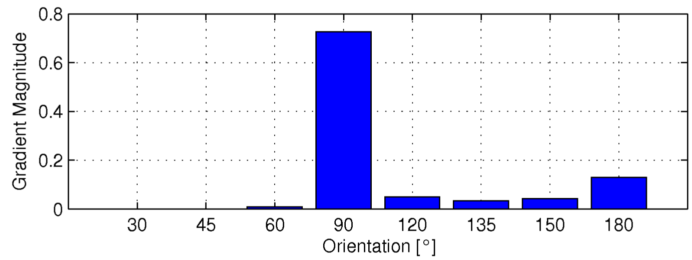
(a) Original frame



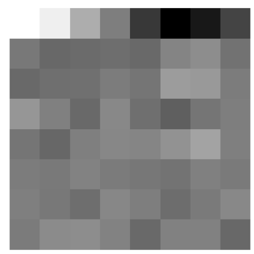
(b) Detected directions

Figure 6: Original frame (a) and overplot (b) of detected directions (green lines).

(a) Residual Block (8×8)

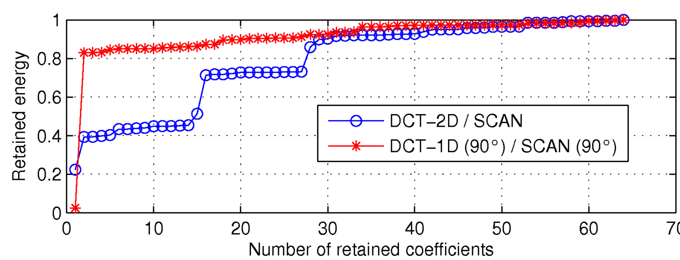


(b) Associated Histogram



(c) 2D-DCT coefficients

(d) 1D-DCT (90°) coefficients



(e) Retained energy of transformed coefficients after scan

Figure 7: Results of integrated solution.

Considering the new coding structure of HEVC, allowing transform sizes of 4×4 up to 32×32, increase in coding time would be much greater, which could be prohibitive for real-time applications.

By using the direction prediction algorithm, despite the many possibilities of transforms, the algorithm previously chooses one directional transform (or conventional 2-D DCT), avoiding RDO calculations and reducing complexity to similar levels when only one transform is applied (assuming that the algorithm itself has negligible complexity). For Kamisli's proposal, estimated reduction in total coding time would be 37.5% (4×4 blocks with 8 possible directions) and 47.4% (8×8 blocks with 16 possible directions).

VI. Conclusion

Recent studies have shown that prediction residuals usually have directional structures that can not be well represented by conventional 2-D DCT. Many directional transforms have been proposed to exploit this directionality, reducing the number of coefficients to fully represent the image and thereby increasing compression efficiency of codecs.

Some of this recently proposed directional transforms use the RDO technique for selection of the direction to be explored. This is very expensive and time-consuming, as all possible options have to be tested and the best option is then chosen, considerably increasing encoder complexity.

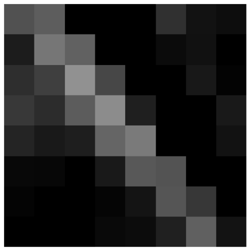
This paper proposes a simple and fast algorithm for estimating predominant directions in residual blocks, previously indicating which directional transform should be applied and avoiding RDO calculations.

Simulations were carried out together with directional transforms proposed in [30], and results showed that directions are correctly identified, and the directional transform indicated by the algorithm performs significantly better than conventional 2-D DCT.

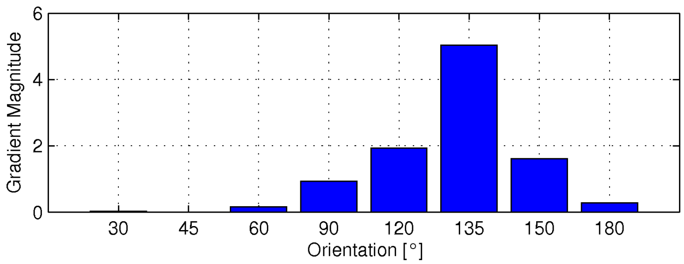
By using this integrated solution, considerably reduction of encoding time (almost 50%) can be achieved, reducing encoder complexity to similar levels when only one transform is applied.

Acknowledgment

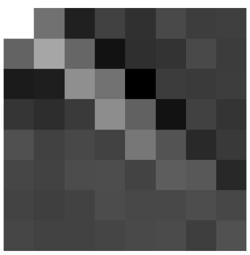
The authors would like to thank the CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*),



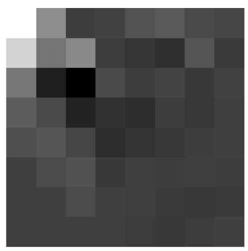
(a) Residual Block (8 × 8)



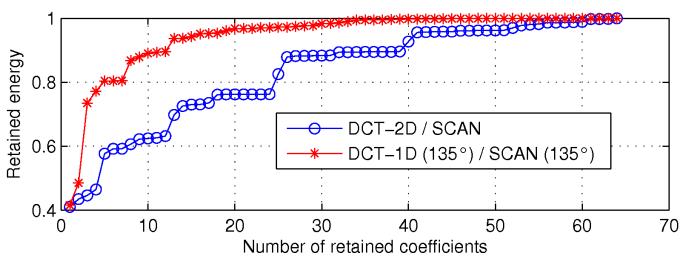
(b) Associated Histogram



(c) 2D-DCT coefficients



(d) 1D-DCT (135°) coefficients



(e) Retained energy of transformed coefficients after scan

Figure 8: Results of integrated solution.

CNPQ (*Conselho Nacional de Desenvolvimento Cientifico e Tecnológico*) and FAPESP (*Fundação de Amparo À Pesquisa do Estado de São Paulo*) programs for the financial support and the academic incentive.

References

[1] "ITU-T JCT-VC website. [online]. Available: http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jctvc.aspx."

[2] *Advanced Video Coding*. ITU-T Rec. H.264 ISO/IEC 14496-10:2009, Mar 2010.

[3] J. Xu, B. Zeng, and F. Wu, "An Overview of Directional Transforms in Image Coding," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 3036–3039, IEEE, 2010.

[4] B. Zeng and J. Fu, "Directional Discrete Cosine Transforms—a New Framework for Image Coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 3, pp. 305–313, 2008.

[5] J. Dong and K. Ngan, "Two-layer Directional Transform for High Performance Video Coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, no. 99, pp. 1–1, 2011.

[6] R. Cohen, S. Klomp, A. Vetro, and H. Sun, "Direction-adaptive Transforms for Coding Prediction Residuals," in *IEEE International Conference on Image Processing*, 2010.

[7] X. Zhao, L. Zhang, S. Ma, and W. Gao, "Video Coding with Rate-Distortion Optimized Transform," *Circuits and Systems for Video Technology, IEEE Transactions on*, no. 99, pp. 1–1, 2012.

[8] G. Sullivan and T. Wiegand, "Rate-distortion Optimization for Video Compression," *Signal Processing Magazine, IEEE*, vol. 15, no. 6, pp. 74–90, 1998.

[9] JCT-VC, "HM6: High Efficiency Video Coding (HEVC) Test Model 6 Encoder Description," Tech. Rep. JCTVC-H1002, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), San José, CA, USA, Feb. 2012.

[10] *Joint Call for Proposals on Video Compression Technology*. ITU-T SG16 Q6 document VCEG-AM91 and ISO/IEC JTC1/SC29/WG11 document N11113, ITU-T SG16 Q6 and ISO/IEC JTC1/SC29/WG11, Jan. 2010.

[11] JCT-VC, "Test Model Under Consideration," Tech. Rep. JCTVC-A205, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), Dresden, Germany, Apr. 2010.

[12] I. Amonou et al., "Description of Video Coding Technology Proposal by France Telecom, NTT, NTT DOCOMO, Panasonic and Technicolor," Tech. Rep. JCTVC-A114, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), Dresden, Germany, Apr. 2010.

[13] M. Winken et al., "Video Coding Technology Proposal by Fraunhofer HHI," Tech. Rep. JCTVC-A116, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), Dresden, Germany, Apr. 2010.

[14] K. Ugur et al., "Video Coding Technology Proposal by Tandberg,Nokia, and Ericsson," Tech. Rep. JCTVC-A119, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), Dresden, Germany, Apr. 2010.

[15] M. Karczewicz et al., "Video Coding Technology Proposal by Qualcomm," Tech. Rep. JCTVC-A121, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), Dresden, Germany, Apr. 2010.

[16] W. J. Han et al., "Video Coding Technology Proposal by Samsung (and BBC)," Tech. Rep. JCTVC-A124, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Col-

laborative Team on Video Coding (JCT-VC), Dresden, Germany, Apr. 2010.

[17] T. Davies et al., "Video Coding Technology Proposal by Samsung (and BBC)," Tech. Rep. JCTVC-A125, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), Dresden, Germany, Apr. 2010.

[18] JCT-VC, "Encoder-side description of HEVC Test Model (HM)," Tech. Rep. JCTVC-C402, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), Guangzou, CN, Oct. 2010.

[19] JCT-VC, "HEVC Reference Software Manual," Tech. Rep. JCTVC-D404, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), Daegu, KR, Jan. 2011.

[20] G. Sullivan and J. Ohm, "Recent Developments in Standardization of High Efficiency Video Coding (HEVC)," in *Proc. SPIE*, vol. 7798, pp. 7798–30, 2010.

[21] T. Wiegand, J. Ohm, G. Sullivan, W. Han, R. Joshi, T. Tan, and K. Ugur, "Special Section on the Joint Call for Proposals on High Efficiency Video Coding (HEVC) Standardization," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 12, pp. 1661–1666, 2010.

[22] W. Han, J. Min, I. Kim, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. Hong, *et al.*, "Improved Video Compression Efficiency Through Flexible Unit Representation and Corresponding Extension of Coding Tools," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 12, pp. 1709–1720, 2010.

[23] J. Han, A. Saxena, and K. Rose, "Towards Jointly Optimal Spatial Prediction and Adaptive Transform in Video/Image Coding," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 726–729, IEEE, 2010.

[24] JCT-VC, "Non-CE 7: Experimental Results for the ROT," Tech. Rep. JCTVC-H0456, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), San Jose, USA, Feb. 2012.

[25] JCT-VC, "CE7: On Secondary Transforms for Intra Prediction Residual," Tech. Rep. JCTVC-G108_v4, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Joint Collaborative Team on Video Coding (JCT-VC), Geneva, CH, Nov. 2011.

[26] W. Chen, C. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," *Communications, IEEE Transactions on*, vol. 25, no. 9, pp. 1004–1009, 1977.

[27] D. Lowe, "Distinctive Image Features from Scale-invariant Keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] F. Pan, X. Lin, S. Rahardja, K. Lim, Z. Li, D. Wu, and S. Wu, "Fast Mode Decision Algorithm for Intraprediction in H.264/AVC Video Coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 7, pp. 813–822, 2005.

[29] W. Jiang, H. Ma, and Y. Chen, "Gradient Based Fast Mode Decision Algorithm for Intra Prediction in HEVC,"

[30] F. Kamisli and J. S. Lim, "1-D Transforms for the Motion Compensation Residual," *Image Processing, IEEE Transactions on*, no. 99, pp. 1036–1046, 2011.

[31] "HEVC ftp test sequences. [online]. available: "ftp://hevc@ftp.tnt.uni-hannover.de/orig-cfp/"."

Gabriel Beltrão is currently a M.Sc. candidate in the Faculty of Electrical and Computer Engineering, at the University of Campinas (UNICAMP), Brazil. He received his B.S degree in Telecommunications Technology at UNICAMP and Electrical Engineering at Universidade Salvador (UNIFACS) in 2007 and 2010 respectively. Currently he is a researcher in the Visual Communications Laboratory on the same University. His research interests include image and video processing, compression and radar signal processing.

Julio León is currently a Ph.D. candidate in the Faculty of Electrical and Computer Engineering, at the University of Campinas (UNICAMP), Brazil. In 2007, he received his B.Sc degree in Electrical Engineering at Universidad Peruana de Ciencias Aplicadas, Lima-Peru and then obtained the M.Sc. degree in Electrical Engineering at UNICAMP in 2011. He also works at the Visual Communications Laboratory (LCV) on the same University and his research interests include image signal processing, radar signal processing and Smart Grid.

Rangel Arthur received the B.S. degree in Electrical Engineering from the State University of São Paulo (UNESP), Ilha Solteira, Brazil, in 1999, and the M.Sc. and the Ph.D. degrees in Electrical Engineering in 2002 and 2007, respectively, from the Faculty of Electrical and Computer Engineering, University of Campinas (UNICAMP), Brazil. Currently he is a Professor at the Telecommunication Division in the Faculty of Technology, UNICAMP, Brazil. His current research interest includes digital television in single frequency network environments, video compressing based on wavelets transform, video codec projects using digital signal processors and FPGA, turbo and LDPC coding and mobile health devices.

Yuzo Iano received his Ph.D. degree in Electrical Engineering from the Faculty of Electrical and Computer Engineering, University of Campinas (UNICAMP), Brazil, in 1986. Currently he is a Full Professor at the Communications Department in UNICAMP. He is also head of the Visual Communications Laboratory (LCV) at the same university. He is responsible for many digital signal processing (sound and image) projects, and his research interests include video and audio communications, digital signal processing, digital TV, middleware and signal transmission.