

# Low Computational Complexity Variable Block Size (VBS) Partitioning for Motion Estimation using the Walsh Hadamard Transform (WHT)

Chanyul Kim and Noel E.O'Connor,

CLARITY: Centre for Sensor Web Technologies, Dublin City University, Ireland

E-mail: dionism@gmail.com;noel.oconnor@dcu.ie

## Abstract

*Variable Block Size (VBS) based motion estimation has been adapted in state of the art video coding, such as H.264/AVC, VC-1. However, a low complexity H.264/AVC encoder cannot take advantage of VBS due to its power consumption requirements. In this paper, we present a VBS partition algorithm based on a binary motion edge map without either initial motion estimation or Rate-Distortion (R-D) optimization for selecting modes. The proposed algorithm uses the Walsh Hadamard Transform (WHT) to create a binary edge map, which provides a computational complexity cost effectiveness compared to other light segmentation methods typically used to detect the required region.*

operations and 2 thresholdings per a  $3 \times 3$  block. Our approach described in this paper requires only 8 additions and 1 thresholding per a  $4 \times 4$  block. Moreover the threshold value should be decided empirically. In [7], a Cellular Non-linear Network (CNN) type segmentation algorithm was used for detecting edge information. They used an edge enhancing low-pass filter to find regions that contain remarkable features, i.e. edges. Both prior works are performed in the pixel domain using various gradient filters, which introduces a heavy burden of computations. Our proposed VBS partition algorithm has two distinguishing features. Firstly, all processing is performed in the WHT domain, making it is easy to predict residual data's characteristics. Secondly, a computational cost effective algorithm compared to other related works to detect features is presented.

## 1. Introduction

In recent years, the VBS motion estimation technique has been widely employed to improve the performance of the Block Matching Algorithm (BMA). In VBS, the block size is varied according to the type of motion. It is known to be very efficient for areas containing complex motions. However, it requires a large number of computational complexity. Therefore the traditional methods to decide VBS perform it after exhaustive motion estimation and R-D optimization. Clearly, it is not suitable for power limited platforms. There have been several attempts to reduce the computational complexity of VBS partitioning recently based on not performing motion estimation in advance. In this scenario, light segmentation of block characteristics is used, which introduce its own complexity. Therefore low complexity segmentation algorithms have become an important requirement for an encoding process on a power or computation constrained platform. In [9], an edge block detection based subsampling method was proposed. They used Robert cross convolution masks to detect if the block is either an "Edge block" or a "Flat block". However their approach requires 15 additions and 16 absolute difference

## 2 Walsh Hadamard Transform (WHT)

Since it is simple and efficient to execute, the WHT has been applied in many fields, such as pattern matching [4], feature recognition [13], wireless communication [3], and image/video compressions [2]. It is attractive due to the simplicity of its implementation and to its properties which are similar to the familiar Discrete Transform (DT). Many DTs have been used in image processing such as Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT) and Discrete Tchebichef Transform (DTT) that has recently comes under the spotlight [6]. However such transforms are often hard to implement in real time in some applications due to their computational complexity of floating operations. Even when fast algorithms exist, their inverse transforms do not match the original which can cause a drift effect in image/video compression. In order to solve these problems, integer algorithms were developed and deployed in recent video standards MPEG-4 PART 10 Advanced Video Coding (H.264/AVC) and VC-1 combined with a quantisation procedure named Integer Cosine Transform (ICT) [10]. Also recently, the Integer Discrete Tchebichef Transform (IDTT) was proposed in [6]. However they fo-

cus on a  $4 \times 4$  or a  $8 \times 8$  limited block size which is not extensible to an arbitrary block. They still introduce computational complexity even though they have multiplier free structure. Although the performance of the WHT is inferior to other DTs on image/video processing, it provides comparable performance on images that show rapid gradient changes [5] and its computational efficiency makes it very attractive image processing directly in the transform domain since the elements of the basis vectors are orthogonal and contain only binary values ( $\pm 1$ ).

The WHT has different kinds of order; “natural order”, “dyadic order”, and “sequency order”. The natural order of the WHT is equivalent to the post-permutation algorithm of the fast Fourier transform (FFT) and dyadic order represents the machine-oriented algorithm of the FFT [11]. On the contrary, sequency order is analogous to frequency in DFT, the row vectors of an Sequency ordered Walsh Hadamard Transform (SWHT) matrix are arranged in the ascending order of sequencies which is useful for image processing given its energy compaction property.

## 2.1 The Properties of the WHT

The lowest-order Walsh Hadamard Matrix (WHM) is of order two.

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (1)$$

Its higher order can be obtained via a recursive method.

$$H_N = \bigotimes_{i=1}^n H_2 = \overbrace{H_2 \otimes \dots \otimes H_2}^n \quad (2)$$

where  $\otimes$  represents the Kronecker product. Let the array  $[i(x, y)]$  represents the intensity samples of an original image over an array of  $N^2$ , ( $N = 2^k$ ). The 2-D Hadamard Transform,  $[F(u, v)]$  is given as

$$[F(u, v)] = H_N[f(x, y)]H_N^T = \frac{1}{N}H_N[f(x, y)]H_N \quad (3)$$

since the WHT has orthogonal, symmetric, and unitary properties;

$$H_N H_N^T = NI, \quad H_N H_N^{-1} = NI, \quad H_N^{-1} H_N^T = NI \quad (4)$$

where  $H_N^T$  and  $H_N^{-1}$  represent a transpose and inverse matrix of  $H_N$  respectively,  $I$  is an identity matrix. Its inverse is expressed as:

$$\begin{aligned} H_N[F(x, y)]H_N^T &= H_N H_N[f(x, y)]H_N^T H_N^T \\ &= N^2[f(x, y)], \end{aligned} \quad (5)$$

$$[f(x, y)] = \frac{1}{N^2}H_N[F(x, y)]H_N^T$$

The WHT has several interesting properties. The most important properties from the standpoint of image coding are dynamic range, conservation of energy, and energy compaction as like other DTs.

- The zero sequency term is a measure of the average brightness of a block.

$$F(0, 0) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \quad (6)$$

The maximum possible value of the zero sequency term is  $N^2 A$ , where  $A$  is the maximum value of  $f(x, y)$ . Therefore, the magnitude of other samples in the WHT is confined to  $\pm N^2 A/2$ .

- A conservation of energy property, called Parserval Theorem, exists between the spatial domain and the Walsh Hadamard domain.

$$\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |f(x, y)|^2 = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |F(u, v)|^2 \quad (7)$$

## 3 Motion Edge Detection Algorithm

In this Section, an algorithm for detection motion edges using lowest order WHT ( $2 \times 2$  block) is presented. A block with motion edges generates more inter prediction error compared to a homogeneous one, which is verified via a mathematical analysis. Then the edge detection algorithm and its results will be discussed.

### 3.1 Prediction Error Analysis of Edge Gradient

Two temporal and spatial intensity functions are defined as  $f_1(x)$ ,  $f_2(x)$  and  $g_1(y)$  respectively. These are continuous image signals sampled by the sensor before discretization shown in Figure 1. The prediction errors are denoted as (8), and its variance ( $E(\epsilon_1^2) + E(\epsilon_2^2)$ ) denotes the total power of prediction errors.

$$\begin{aligned} \epsilon_1 &= f_1(t) - f_1(t-1) \\ \epsilon_2 &= f_2(t) - f_2(t-1) \end{aligned} \quad (8)$$

The subtraction of each prediction error is denoted in (9), where  $\Delta_{d1}$ ,  $\Delta_{d2}$  represent temporal displacement error.

$$\begin{aligned} \epsilon_1 - \epsilon_2 &= f_1(t) - f_1(t-1) - f_2(t) + f_2(t-1) \\ &= f_1(t) - \Delta_{d1} \times f_1(t) - f_2(t) + \Delta_{d2} \times f_2(t) \end{aligned} \quad (9)$$

The temporal displacement error of two point is almost the same because all motion vectors in the block are invariant. Therefore, (9) is rewritten as (10).

$$\epsilon_1 - \epsilon_2 \simeq (1 - \Delta_{d1})(f_1(t) - f_2(t)) \quad (10)$$

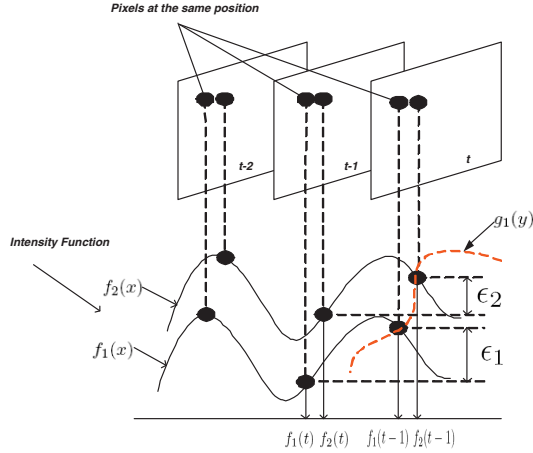


Figure 1: Inter prediction error analysis

In case of infinite sampling periods, it is simplified in (11) using the differential of the function,  $g_1'(y)$ , at  $y = s$  at time  $t$ .

$$\lim_{t \rightarrow 0} (\epsilon_1 - \epsilon_2) = (1 - \Delta_{d1}) \times g_1'(s) \quad (11)$$

The temporal displacement error  $\Delta_{d1}$  is a random variable with zero mean,  $\Delta_{d1} \in [-search, +search]$ . It is assumed that prediction error  $\epsilon_1$  and  $\epsilon_2$  are a memory less stationary Gaussian source of zero means and variances ( $\sigma_1^2, \sigma_2^2$ ), the total power of prediction error ( $\sigma^2$ ) is expressed as in (12).

$$\begin{aligned} \sigma^2 &= \sigma_1^2 + \sigma_2^2 = E((\epsilon_1 - \epsilon_2)^2) = E((\epsilon_1 + \epsilon_2)^2) \\ &= E((1 - \Delta_{d1})^2) \times (g_1'(s))^2 \\ &= \underbrace{(1 + E(\Delta_{d1}^2))}_{\text{motion}} \times \underbrace{(g_1'(s))^2}_{\text{edge}} \end{aligned} \quad (12)$$

Therefore,  $\sigma^2$  should be linear to  $(g_1'(s))^2$ , which means that the average prediction error magnitude is mainly determined by the edge gradient. When a picture contains a lot of edge information, its prediction error from previous frame will be significant. Therefore, in this case, VBS should be considered. On the contrary, when a block is homogeneous, the redundant computational complexity is removed without coding quality degradation. As a result, blocks with edge information and motion (named motion edge in the remainder) need to be detected before the encoding process to reduce inter prediction error and redundant computational complexity.

### 3.2 Motion Edge Detection

It is well known the human visual system is very sensitive to edge information. From (12), the prediction errors of blocks located on edges of objects are not estimated accurately. Moreover, it causes worst prediction error along

the motion edges. Therefore, motion edges of the frame are detected for partitioned VBS. In a nutshell, the procedure of the motion edge detection is as follows;

1. A  $2 \times 2$  block is selected from the top and left position of the frame, and its  $2 \times 2$  WHT coefficients are calculated.
2. From (7), the total power of a  $2 \times 2$  block is conservative in the transform domain. Therefore, the edge information can be obtained from the statistics of non zero sequency terms ( $F(1, 0), F(0, 1), F(1, 1)$ ).
3. The good approximation to the distribution of non zero sequency term is a variance, however its computational complexity is so high that it cannot be applied for complexity constrained systems. Instead of obtaining a variance, the maximum value of non zero sequency term is useful to obtain the similar property as a variance since the dynamic range in the transformed coefficients of non zero sequency term is limited by the zero sequency term shown in (6). When the condition of (13), this block is considered to contain an edge.

$$\max_{(u,v) \neq (0,0)} F(u, v) \geq \tau \quad (13)$$

where  $\tau$  is threshold value, which is insensitive to the image characteristics since only a small block ( $2 \times 2$ ) is used for thresholds.

4. Motion edges are obtained in the same fashion with (13), which is slightly modified as follows;

$$\max_{(u,v) \neq (0,0)} |F(u, v)_t - F(u, v)_{t-1}| \geq \tau \quad (14)$$

where  $F(u, v)_t$  and  $F(u, v)_{t-1}$  represents the transformed block at the same location both in the current and the previous frame.

5. When a  $2 \times 2$  block is classified as a motion edge, the block is mapped to one pixel, so a 4:1 down sampled motion edge frame is obtained from the original image without any further processing introduced. It is computational complexity efficient when VBS directly performs on the down sampled binary motion edge frame.

Figure 2 illustrates a comparison of the result of the proposed edge detection (b), using the Canny edge detection [1](d), which has a very accurate edge detection performance. The proposed method shows more edge pixel compared to (d), since Canny edge could be obtained by detecting zero crossing, which detects a single edge line when it performs on the boundary of object has texture information. The results of proposed method here shows comparable edges over all image resolutions (a), ( $352 \times 288 \rightarrow$

1280 × 720). The detection of a motion edge follows the same approach as displayed in (c). Note that an edge and a motion edge image are 4: 1 subsampled, (b) and (c) are intentionally displayed as the same size for clear distinction.



Figure 2: Motion edge and edge detection results (a) original image (all images are the 46<sup>th</sup> frame ); (b) 4:1 down sampled binary edge image; (c) 4:1 down sampled binary motion edge image; (d) binary edge image captured by Canny edge detection, double thresholds value are [100, 180]

Figure 3 shows the effect of threshold value,  $\tau$ . As  $\tau$  is increase, weak edge pixels move to the background pixels. Therefore, when high Quantisation Parameter (QP) is applied on the image, the motion edge detection on reconstructed image is equivalent to increasing  $\tau$ , which is important concept for the video encoder because we can estimate the reconstructed image not to perform encoding process controlling the threshold value,  $\tau$ . The relationship between  $\tau$  and QP is driven and discussed in details in Chapter 4.1.

In terms of computational complexity, Canny edge detection requires several steps; 1) smoothness by applying a Gaussian filter, 2) finding gradients for each direction, 3) double thresholds. A Sobel operator is used as a tool for finding the gradient, it requires 6 additions and 4 shift operations for every three pixels. On the contrary, the proposed method requires 8 additions for every four pixels. Table 1 shows computational complexity of each method. The proposed one achieve computation saving compared to Canny edge by a factor three. Moreover, when we apply to High-Definition (HD) sequence, it only requires 6 ~ 7ms per a frame to detect motion edges. In case of highly complex edge such as “Mobile”, the computational complexity is

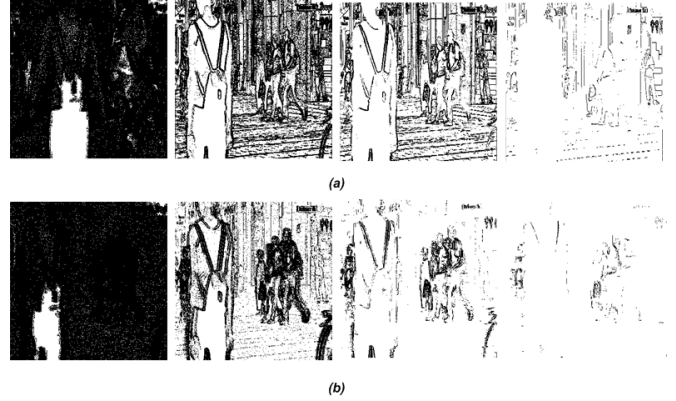


Figure 3: The effect of threshold value  $\tau$ ; (a) binary edge image (b) binary motion edge image; from the left  $\tau$  is 0, 5, 10, 40 at Pedestrian sequence 46<sup>th</sup> frame with 720x576

more reduced by a factor 8. Note that all tests are performed on an Intel Core(TM)2 Duo 3.0GHz with 2GB RAM using Window XP version 2002 with service pack 2 written in ANSI C++.

Table 1: Comparison of execution time of edge or motion edge detection

Unit ( $10 * \frac{ms}{Frame}$ )	Edge	Motion Edge	Canny Edge
Mother (352 × 288)	7.00	8.47	21.21 )
Mobile(352 × 288)	5.68	6.97	41.05
Pedestrian(720 × 576)	27.45	33.19	84.21
Pedestrian(1280 × 720)	63.20	77.02	152.35

## 4 VBS Partitioning for Motion Estimation

In this section a VBS partitioning algorithm for motion estimation is presented using the approach in the previous section. The relationship between threshold value for generating a motion edge image and QP is obtained by simple mathematical analysis. Then the VBS partitioning algorithm is explained in details and its results presented.

### 4.1 Relationship between threshold ( $\tau$ ) and QP

A memory less Laplacian source with zero mean may provide the governing distribution for non-DC DCT or high-frequency wavelet transform coefficients [12, 14]. The characteristic of WHT is similar to that of other DTs. Suppose that  $2 \times 2$  non zero sequency WHT coefficients’ residues, which is used for detecting a motion edge image,

follow a zero mean Laplace distribution, i.e.,

$$p_{lap}(x) = \frac{\Lambda}{2} e^{-\Lambda|x|}, \quad \Lambda = \frac{\sqrt{2}}{\sigma} \quad (15)$$

where  $x$  and  $\sigma$  represent the WHT residues and their standard deviation respectively. For a given QP, the distortion is obtained as following.

$$D(Q) = 2 \times \left( \int_0^{\frac{Q}{2}} x^2 p_{lap}(x) dx \right) + 2 \times \sum_{i=1}^{\infty} \left( \int_{i-\frac{Q}{2}}^{i+\frac{Q}{2}} (x-iQ) p_{lap}(x) dx \right) \quad (16)$$

From [8, 15], closed form of D can be derived as

$$D(Q) = 1/2 \left( \sqrt{2}Q e^{\frac{\sqrt{2}Q}{\sigma}} \left( 2 - \frac{\sqrt{2}Q}{\sigma} \right) \sigma^{-1} + 2 - 2 e^{\frac{\sqrt{2}Q}{\sigma}} \right) \times \sigma^2 \left( 1 - e^{\frac{\sqrt{2}Q}{\sigma}} \right)^{-1} \quad (17)$$

Figure 4 shows the distortion against transformed coefficients variance  $\sigma^2$ . For larger  $\sigma^2$ , the distortion is linear to  $Q^2$ . In our case, the target of  $\sigma^2$  is a large value to decide edge information. So, the distortion can be rewritten for a

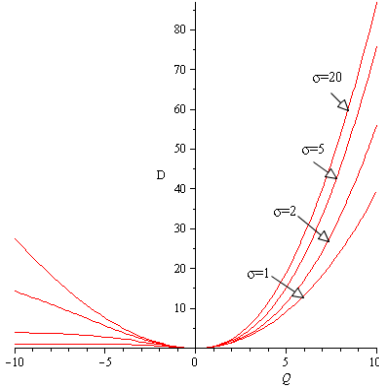


Figure 4:  $D(Q)$  vs variable  $\sigma$ ; the relation  $D(Q) \cong kQ^2$  is obtained for a large  $\sigma$

large  $\sigma$  as

$$D(Q) \cong kQ^2 \quad (18)$$

The distortion is constant for a given QP, therefore (19) is obtained from (12) and (18) as,

$$(1 + E(\Delta_{d1})^2) \times (g'_1(s))^2 \cong kQ^2 \quad (19)$$

Assuming that the displacement error  $E(\Delta_{d1})^2$  is negligible, the edge gradients is also proportional to threshold value,  $\tau$ . Therefore the threshold value is also linear to QP:

$$\tau \cong (\beta \times Q) \quad (20)$$

where  $\beta = \sqrt{k}$ .

(20) shows the fact that the variation of QP is similar to that of  $\tau$ . Our proposed VBS partitioning algorithm does not perform the encoding processing; we need to estimate the output of quantised signal which is usually obtained after the encoding processing. However, it is hard to understand the behavior of QP in the pixel domain because it works on the transformed coefficients. The method presented here works in the transform domain, it enables the encoder to obtain the similar image signal after QP by adjusting  $\tau$  not to encode directly.

## 4.2 VBS partition algorithm

The VBS partition algorithm makes use of a binary motion image, the procedure is as follows;

1. A  $16 \times 16$  macro block is chosen from the top, left position of the frame.
2. When macro block area of the motion edge image has a motion edge as per (1), this block is partitioned as  $8 \times 8$  or  $4 \times 4$  block size depending on edge pixel location.
3. Check the position in motion edge has a value of 1, then this block is partitioned into smaller blocks by recursive method to a  $4 \times 4$  block size.
4. Perform above procedure on the rest of macro blocks with raster scan order.

From (20), the threshold value,  $\tau$  should be applied because a binary motion edge map is obtained before reconstructing an image. The proposed method operates on the WHT domain, so it is easy to estimate the reconstructed image characteristic – usually residual data are coded by DCT which has similar characteristics to WHT.

Figure 5 shows the R-D performance of the proposed method compared to JM11.0 with full search motion estimation. The proposed approach can greatly reduce motion estimation time and achieves almost the same R-D performance as the original encoder. As  $\tau$  is getting large, the blocks have weak motion edges go to homogenous blocks. Clearly, it enable us to control computational complexity automatically not to consider where weak motion edge is. Segmented VBSs are shown in Figure 6 at various  $\tau$ .

## 5 Conclusion

Despite the fact that there are a large number of VBS partitioning algorithms presented by various researches, a few promising techniques can be identified as the potentially useful approaches for computational complexity. In this paper, the fundamental features of the WHT are overviewed, that is, its energy conservation, dynamic range and energy

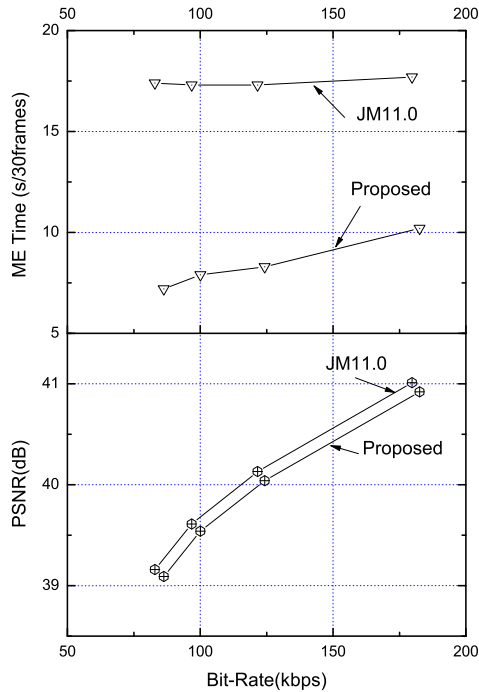


Figure 5: R-D performance comparison on Mother and Daughter sequence with  $352 \times 288$  at  $\tau = 15$

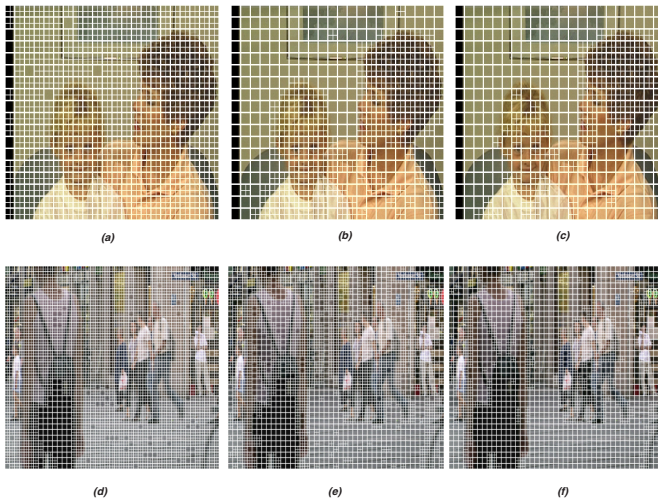


Figure 6: VBS partitioned results at various threshold value;  $\tau$ , (a) 5, (b) 10, (c) 20; (a)(b)(c) “Mother and Daughter” at  $352 \times 288$ , (d)(e)(f) “Pedestrian” at  $720 \times 576$

compactness. Using graphical approach, the inter prediction errors are mainly originate from the spatial gradients and motion variation. So we focus on detecting motion edges. Binary motion edge detection algorithms are presented in the WHT domain. The results shows that it is computational cost effective compared to other edge detection algorithms such as Canny and Sobel operator. Moreover, the relationship between threshold value ( $\tau$ ) and QP is suggested. Finally, VBS algorithms based on the motion edge detection is presented, its results show that it can be used as a basic tool for motion estimation without exhaustive modes decision via R-D optimization. In the future work, we need to investigate the combination algorithm between VBS and fast motion estimation especially performed on WHT.

## References

- [1] J.F. Canny. A computational approach to edge detection. *IEEE Trans Patten Analysis and Machine Intelligence*, 8:679–698, 1986.
- [2] R. Costantini, J. Bracamonte, G. Ramponi, J-L. Nagel, M. Ansorge, and F. Pellandini. A low-complexity video coder based on the discrete walsh hadamard transform. In *EUPSICO 2000 : European signal processing conference*, pages 1217–1220, 2000.
- [3] Zhenghui Gu, Shoulie Xie, and S. Rahardja. Unified complex hadamard transform sequences for multi-carrier CDMA systems. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 3, pages 1514–1517, May 2004.
- [4] Y. Hel-Or and H. Hel-Or. Real-time pattern matching using projection kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1430–1445, September 2005.
- [5] O Hunt and R. Mukundan. A comparison of discrete orthogonal basis functions for image compression. In *Image and Vision Computing New Zealand (IVCNZ-2004)*, pages 53–58, 2004.
- [6] S. Ishwar, P. K. Meher, and M. N. S. Swamy. Discrete tchebichef transform-a fast 4x4 algorithm and its application in image/video compression. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 260–263, Seattle, WA, May 2008.
- [7] Lauri Koskinen, Ari Paasio, and Kari Halonen. Cnn-type algorithms for h.264 variable block-size partitioning. *Image Commun.*, 22(9):797–808, 2007. ISSN 0923-5965.
- [8] Xiang Li, N. Oertel, A. Hutter, and A. Kaup. Laplace distribution based lagrangian rate distortion optimization for hybrid video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(2):193–205, February 2009.
- [9] Qin LIU, Yiqing HUANG, Satoshi GOTO, and Takeshi IKENAGA. Edge block detection and motion vector information based fast vbsme algorithm. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences Advance Access*, E91(A), Aug 2008.
- [10] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky. Low-complexity transform and quantization in h.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):598–603, July 2003.
- [11] M.J. Corinthis. A time-series analyzer. *Computer Processing in Communication*, pages 47–69, Apr 1969.
- [12] Laplacian Source Nasir. Simulation of the rate-distortion behaviour of a memoryless, 2002.
- [13] Wei-Hau Pan, Shou-Der Wei, , and Shang-Hong Lai. Efficient ncc-based image matching in walsh-hadamard domain. *Lecture Notes in Computer Science*, pages 468–480, Oct 2008.
- [14] Stephen Smoot and Lawrence A. Rowe. Laplacian model for ac dct terms in image and video coding. In *Proceedings of the 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video Table of Contents, MontereyCA*, pages 60–69, 1996.
- [15] Long Xu, Xiangyang Ji, Wen Gao, and Debin Zhao. Laplacian distribution model (LDM) for rate control in video coding. In *Advances in multimedia information processing PCM2007*, pages 11–14, Hong Kong, 2007.