PREDICTION SIGNAL AIDED SPATIALLY VARYING TRANSFORM

Cixun Zhang[*], *Kemal Ugur*[§], *Jani Lainema*[§], *Antti Hallapuro*[§], *Moncef Gabbouj*[*]

[*]Tampere University of Technology, Tampere, Finland
[§]Nokia Research Center, Tampere, Finland

ABSTRACT

Spatially Varying Transform (SVT) is a technique introduced earlier to improve the coding efficiency of video coders [1][2]. SVT allows the position of the transform block within the macroblock to vary in order to better localize the underlying residual signal. The coding gains of SVT come with increased encoding complexity due to the additional need in the encoder to search for the best Location Parameter (LP) which indicates the position of the transform. In this paper, a new technique called Prediction Signal Aided Spatially Varying Transform (PSASVT) is proposed that utilizes the gradient of prediction signal to eliminate the unlikely LPs. As the number of candidate LPs is reduced, a smaller number of LPs are searched by encoder, which reduces the encoding complexity. In addition, less overhead bits are needed to code the selected LP and thus the coding efficiency can be improved. Experimental results show that the number of LPs to be tested in RDO is reduced on average by more than 20%. This reduction in encoding complexity is achieved with a slight increase in coding efficiency, as the number of candidate LPs is reduced. The decoding complexity increase is only a little.

***Index Terms*—** Video coding, Transform, Spatially Varying Transform (SVT)

1. INTRODUCTION

Nowadays, as display resolutions and available bandwidth/storage increase rapidly, High-Definition (HD) video is becoming more popular and commonly used. This makes the implementation of video codecs more challenging, especially in resource constrained applications, such as mobile video services (mobile TV, video telephony etc) and handheld consumer electronics (camcorders, digital still cameras etc). To better satisfy the requirements of increased usage of HD video in resource constrained applications, two key issues should be addressed: coding efficiency and implementation complexity. In [1][2], the concept of Spatially Varying Transform is introduced to improve the coding efficiency of video coders. The motivations leading to design of SVT are two-fold:

1. The block based transform design in most existing video coding standards does not adapt the underlying transform to the structure of the prediction error to be coded. This leads to inefficiency in the coding efficiency.
2. Coding the entire prediction error signal may not be the best in terms of rate distortion tradeoff because the prediction error signal may contain noise which contributes little to quality but requires additional bits to code.

Both of these two factors contribute to the coding efficiency improvement achieved by SVT [2].

The basic idea of SVT is that the transform coding is not restricted inside regular block boundaries but is adjusted to the characteristics of the prediction error. With this flexibility, coding efficiency improvement can be achieved by selecting and coding the best portion of the prediction error in terms of rate distortion tradeoff. Generally, this can be done by searching inside a certain residual region after intra prediction or motion compensation, for a sub-region and only coding this sub-region. Finally, the Location Parameter (LP) indicating the position of the sub-region inside the region is coded into the bitstream if necessary. This approach leads however to increased encoding complexity because the encoder needs to search for the best LP among a number of candidate LPs. This added encoding complexity is problematic for real-time encoding use-cases on resource constrained devices.

In this paper, a new technique called Prediction Signal Aided Spatially Varying Transform (PSASVT) is proposed, which enables reducing the number of candidate LPs based on the prediction signal and thus also reducing the encoding complexity. In addition, the overhead bits for transmitting the index of the selected LP are reduced, as there are a smaller number of candidate LPs. Experimental results show that the number of LPs to be tested in RDO is reduced on average by more than 20%. Furthermore, this reduction in encoding complexity is achieved with a slight increase in coding efficiency and only a little decoding complexity increase.

The paper is organized as follows. A brief review of SVT is presented in section 2. Section 3 introduces PSASVT and presents a detailed explanation of its design.

978-1-61284-350-6/11/$26.00 ©2011 IEEE

Experimental results are given in section 4 and section 5 concludes the paper.

2. SPATIALLY VARYING TRANSFORMS

Transform coding is widely used in video coding standards to decorrelate the prediction error and achieve increased compression rates. Typically, transform coding is applied to prediction error at fixed locations. However, this has several drawbacks that may hurt the coding efficiency and decrease visual quality. First of all, if the localized prediction error inside the fixed block boundaries has a structure that is not suitable for the underlying transform, many high frequency coefficients will be generated in the transform domain and they need many bits to code. This leads to inefficiency in the coding efficiency. Moreover, notorious visual artifacts such as ringing may appear when these high frequency coefficients get quantized.

In [1][2], SVT is proposed to reduce these drawbacks of transform coding. The main idea of SVT is that the transform coding is not restricted to be aligned with the traditional grid of block boundaries, but instead can be applied at any location according to the characteristics of the prediction error. With this flexibility, coding efficiency improvement can be achieved by selecting and coding the best portion of the prediction error in terms of rate distortion tradeoff. Generally this can be done as follows. Only a sub-region is coded in a certain residual region after intra/inter prediction. The sub-region is found by searching inside the region according to a certain criterion. Finally, the location parameter (LP) indicating the position of the selected sub-region inside the region is coded into the bitstream if necessary.

For 8×8 SVT, as shown in Fig. 1, the location of the selected 8×8 block inside the current macroblock can be denoted by (Δx, Δy) which can be selected from the set $\Phi_{8\times8}=\{(\Delta x, \Delta y), \Delta x, \Delta y=0,\ldots,8\}$. There are in total 81 candidates for 8×8 SVT. To reduce complexity, it is suggested to use a reduced set $\Phi_{8\times8}'=\{(\Delta x, \Delta y), \Delta x=0,\ldots,8, \Delta y=0; \Delta x=0,\ldots,8, \Delta y=8; \Delta x=0, \Delta y=1,\ldots,7; \Delta x=8, \Delta y=1,\ldots,7\}$ instead. For 16×4 SVT and 4×16 SVT, as shown in Fig. 2, the locations of the selected 16×4 and 4×16 block inside the current macroblock can be denoted as Δy and Δx, respectively, which can be selected from the set $\Phi_{16\times4}=\{\Delta y, \Delta y=0,\ldots,12\}$ and $\Phi_{4\times16}=\{\Delta x, \Delta x=0,\ldots,12\}$ respectively. In total there are 26 candidates for 16×4 SVT and 4×16 SVT. Rate Distortion Optimization (RDO) is used to select the best LP for SVT and determine whether SVT is used for coding each macroblock.

For the selected SVT block, corresponding block-size transform is applied accordingly. Generally, the separable forward and inverse 2-D transform of a 2-D signal can be written as

$$C = T_v \cdot X \cdot T_h^T , \tag{1}$$

$$X_r = T_v^T \cdot C \cdot T_h \tag{2}$$

respectively, where **X** denotes a matrix representing *M*×*N* pixel block of *N* pixels horizontally and *M* pixels vertically, **C** is the transform coefficient matrix, and $\mathbf{X_r}$ denotes a matrix representing reconstructed signal block. $\mathbf{T_v}$ and $\mathbf{T_h}$ are the *M*×*M* and *N*×*N* transform kernels in vertical and horizontal direction, respectively. The superscript T denotes matrix transposition. Traditional zig-zag scan is used to represent the transform coefficients as input symbols to the entropy coding.

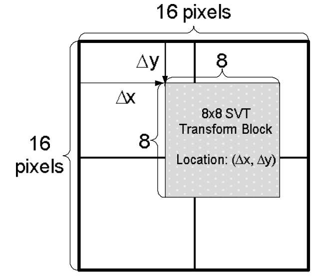

Fig.1 Illustration of 8×8 spatially varying transform

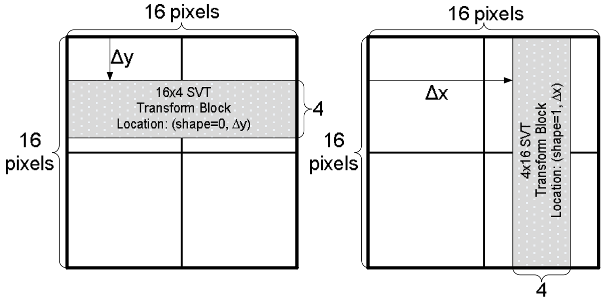

Fig.2 Illustration of 16×4 and 4×16 spatially varying transform

It is noted that even though the motion estimation, sub-macroblock partition decision process for the macroblocks that use SVT are not changed [1][2], the encoding complexity of SVT is higher due to the brute force search process in RDO. For example, there are a total of 58 candidate LPs for one macroblock mode and RDO needs to be conducted for each candidate LP to select the best one. Note that typically one needs to conduct transform, quantization, entropy coding, inverse transform and inverse quantization etc to calculate the RD cost accurately and this complexity is high. This is problematic especially for resource constrained applications.

3. PREDICTION SIGNAL AIDED SPATIALLY VARYING TRANSFORMS

As mentioned above, the encoding complexity for SVT is increased because the encoder needs to search for the best LP among a number of candidate LPs. The basic idea to reduce the encoding complexity of SVT is to reduce the number of candidate LPs tested in RDO. In this paper, we propose to select candidate LPs based on the prediction signal. The motivations leading to this design are two-fold:

1. Statistics show that normally the selected SVT block locates at positions of the macroblock where the prediction error has higher magnitude [1][2]. These positions are probably boundary positions [3][10]. One example is illustrated in Fig. 3 below [2].

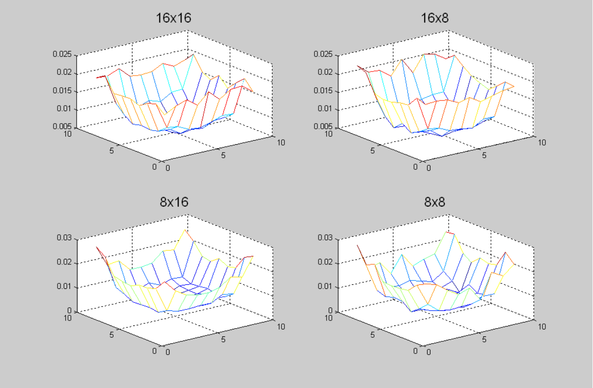

Fig.3 Distribution of (Δx, Δy) of 8×8 SVT for BigShips sequence.

2. The gradient of the prediction signal is assumed to be positively correlated with the magnitude of the prediction error, i.e., a larger gradient indicates a higher prediction error magnitude. In other words, it is probable that when the gradient of the prediction signal is high, the texture of the original signal is more complex and hard to predict well thus the prediction error has a relatively high magnitude. A similar assumption is also used in [4].

The proposed algorithm PSASVT is summarized below:

Step 1: Calculate the gradient map of the prediction signal of the current macroblock using Sobel operator, as follows.

$$G(x,y)=\left|\begin{array}{l}S(x-1,y-1)+2S(x,y-1)+S(x+1,y-1)\\ -S(x-1,y+1)-2S(x,y+1)-S(x+1,y+1)\end{array}\right| +\left|\begin{array}{l}S(x-1,y-1)+2S(x-1,y)+S(x-1,y+1)\\ -S(x+1,y-1)-2S(x+1,y)-S(x+1,y+1)\end{array}\right| \quad (3)$$

where $G(x,y)$ is the gradient value and $S(x,y)$ is the prediction signal value at position (x,y).

Step 2: If the gradient map is all-zero map, then in this case we assume the prediction quality is very good and thus we only try 8 corner positions at which the prediction error is assumed to be larger than other positions, i.e., the set $\Phi_c = \Phi_{8\times8}'' + \Phi_{16\times4}' + \Phi_{4\times16}'$, where $\Phi_{8\times8}'' = \{(\Delta x, \Delta y), \Delta x=0, \Delta y=0; \Delta x=8, \Delta y=0; \Delta x=0, \Delta y=8; \Delta x=8, \Delta y=8;\}$; $\Phi_{16\times4}' = \{\Delta y, \Delta y=0, 12\}$; $\Phi_{4\times16}' = \{\Delta x, \Delta x=0, 12\}$; then goto step 5.

Step 3: Otherwise, for each candidate LP, calculate the sum of gradient (denoted by a variable *SoG*) of the prediction signal inside the SVT block at that position.

Step 4: Select some candidate LPs to test in RDO following these two sub-steps:

(a): Calculate a threshold SoG_t as follows:

$$\mathrm{SoG_t} = (\mathrm{SoG_{max}}+\mathrm{SoG_{min}}+1) >> 1; \quad (4)$$

where SoG_{max} and SoG_{min} are the maximum and minimum value of *SoG* among all the candidate LPs, respectively. Generally, the calculation of the threshold SoG_t can be dependent on the statistics of *SoG* of all the candidate LPs, and/or other characteristics of the current (and neighboring) macroblock. A larger threshold SoG_t can reduce more candidate LPs tested in RDO, but on the other hand may degrade the coding efficiency because the selected candidate LPs may not be among the best ones in terms of rate distortion tradeoff, and vice versa. Eq. (4) for calculating the threshold SoG_t is derived according to our experience which shows a good tradeoff between performance and complexity.

(b): A candidate LP is selected to be tested if *SoG* for this candidate LP is larger than or equal to SoG_t.

Besides the above method we also tried the following variations to select the candidate LPs to test and the results are given in section 4 for comparison.

(i) We calculate the average value of *SoG* of all the candidate LPs as the threshold:

$$SoG_t = \sum_{i=1}^{N} SoG_i / N \quad (5)$$

where N is the number of all the candidate LPs. We then test the candidate LPs whose *SoG* is larger than or equal to SoG_t.

(ii) We use the median value of *SoG* of all the candidate LPs as the threshold SoG_t and test the candidate LPs whose *SoG* is larger than or equal to SoG_t.

(iii) We first sort the *SoG* of all the candidate LPs into non-increasing order and calculate the difference of the neighboring *SoG* in the ordered list. Then search the largest difference from the beginning of the ordered list which for example is in the i-th ($1 \le i \le N$, where N is the number of all the candidate LPs) position of the list. Set the threshold SoG_t to be *SoG* in the i-th position of the ordered list. In other words, we use the largest difference of ordered *SoG* as a criterion to derive the threshold SoG_t.

Step 5. End

An example of the PSASVT algorithm is illustrated in Fig. 4 below:

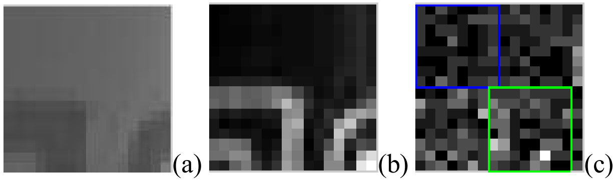

Fig.4 (a) the prediction signal; (b) the gradient map; (c) the absolute prediction error signal; as an example, the SVT block marked in blue has SoG_{min} and will not be checked in RDO and the SVT block marked in green has SoG_{max} and will be checked in RDO.

We note that a more general way to decide on the SVT positions would be to use an estimated RD cost of the whole macroblock for each candidate LP from the prediction signal. Generally this can be done by minimizing the following:

$$J = D + \lambda \cdot R \tag{6}$$

where D is the estimated distortion which can be estimated from the gradient outside the SVT block and R is the estimated rate which can be obtained from the gradient inside the SVT block. This generalization method is under further study, however, according to our experiments, simply calculating the sum of gradient of the prediction signal, as in our current implementation, a good tradeoff between performance and complexity could be achieved.

4. EXPERIMENTAL RESULTS

We implemented the proposed idea PSASVT on Tandberg, Ericsson and Nokia test model (TENTM) [5][6] in order to evaluate its effectiveness. TENTM was a joint proposal to the High-Efficiency Video Coding (HEVC) standardization effort, which achieves similar visual quality measured using Mean Opinion Score (MOS) to H.264/AVC High Profile anchors, and has a better tradeoff between complexity and coding efficiency than H.264/AVC Baseline Profile [5][6]. Because of this, it is selected as the platform in our experiments. The important coding parameters are as follows:

- Quad-tree based coding structure with a support for macroblocks of size 64×64, 32×32 and 16×16 pixels;
- Frame structure is IPPP;
- QP_I=11, 16, 21, 26, QP_P=QP_I+2;
- Low complexity variable length coding scheme with improved context adaptation [5][6];
- The proposed algorithm is only used for 16×16 motion compensation partition. For other motion compensation partitions, a previously developed algorithm of selecting available candidate LP based on motion difference [1][7] is used.

The anchor is the TENTM when SVT is turned off. We measure the average bitrate reduction (ΔBD-RATE) compared to the anchor according to Bjontegaard metric [8], recommended by the VCEG, which is an average difference between two RD curves. The set of sequences tested in this paper corresponds to the Joint Collaborative Team on Video Coding (JCT-VC) test set [9] extended by 6 additional sequences. The results are shown in Table 1 below.

We have tested four variations of the proposed algorithm, namely: (i) PSASVT_MID: use (4) to calculate the threshold SoG_t; (ii) PSASVT_AVG: use (5) to calculate the threshold SoG_t; (iii) PSASVT_MED: use the method as explained in step 3 (ii) of the algorithm in section 3 to calculate the threshold SoG_t; (iv) PSASVT_DIFF: use the method as explained in step 3 (iii) of the algorithm in section 3 to calculate the threshold SoG_t. From the experimental results, we can see that encoding complexity reduction and slight coding efficiency improvement are achieved for all cases. In a general sense, PSASVT_MID performs better than PSASVT_AVG and PSASVT_MED, and can reduce on average 21.70% of the candidate LPs tested in RDO, while also achieves on average 0.18% bitrate reduction. PSASVT_DIFF can reduce most of the candidate LPs tested in RDO (on average 24.46%) among all the cases, while achieves slightly less bitrate reduction (on average 0.13%). The complexity of PSASVT_DIFF is higher than others mainly because it needs to sort the SoG of all the candidate LPs into non-increasing order and calculate the difference of the neighboring SoG in the ordered list. On the contrary, the complexity of PSASVT_MID could be (one of) the lowest among all cases.

For all the cases, the reduction in encoding complexity and slight coding efficiency improvement are achieved with only a little complexity increase in decoder. The decoding complexity increase is mainly due to the calculation of the gradient map of the macroblock and the sum of gradient of the prediction signal inside the SVT block for each candidate LP. This needs only to be conducted when 16×16 motion compensation partition and SVT are used, which is typically only a small percent (less than 5% on average) of all the macroblocks in a sequence. The percentages of macroblocks coded in SVT for different sequences in each case for all partitions and for only 16×16 partition are given in Table 2.

5. CONCLUSIONS

Spatially Varying Transform (SVT) is a technique introduced earlier to improve the coding efficiency of video coders [1][2]. In SVT, the position of the transform block within the macroblock, which is indicated by Location Parameter (LP), can be varied in order to better localize the underlying prediction error signal. The coding efficiency of SVT comes with an increase in encoding complexity due to the need to search for the best candidate LP among a number of candidate LPs. In this paper, a new technique called Prediction Signal Aided Spatially Varying Transform (PSASVT) is proposed, which enables reducing the number of candidate LPs based on the prediction signal. Experimental results show that PSASVT can reduce on average more than

20% of the candidate LPs tested in RDO, while also achieving slight increase in coding efficiency, because less overhead bits are needed to code the selected LP. Moreover, the bitrate reduction and the encoder complexity reduction are achieved with only a little complexity increase in decoder.

6. REFERENCES

[1] C. Zhang, K. Ugur, J. Lainema A. Hallapuro and M. Gabbouj, “Video Coding Using Spatially Varying Transform”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, Issue. 2, pp. 127-140, Feb. 2011.

[2] C. Zhang, K. Ugur, J. Lainema and M. Gabbouj, “Video Coding Using Spatially Varying Transform,” in Advances in Image and Video Technology, Lecture Notes in Computer Science, vol. 5414, Springer 2009.

[3] M. T. Orchard and G. J. Sullivan, “Overlapped block motion compensation: An estimation-theoretic approach,” IEEE Trans. Image Processing, vol. 3, pp. 693-699, Sept. 1994.

[4] H. Schiller, “Prediction signal controlled scans for improved motion compensated video coding”, Electronics Letters, Vol. 29, Issue. 5, pp. 433-435, 1993.

[5] K. Ugur, K.R. Andersson, A. Fuldseth, et al, “High Performance, Low Complexity Video Coding and the Emerging HEVC Coding Standard”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 20, Issue. 12, pp. 1688-1697, Dec. 2010.

[6] K. Ugur, K.R. Andersson, A. Fuldseth, “Video coding technology proposal by Tandberg, Nokia, and Ericsson”, JCTVC-A119, Dresden, Germany, 15-23 April, 2010.

[7] C. Zhang, K. Ugur, J. Lainema, A. Hallapuro, and M. Gabbouj, "Low Complexity Algorithm for Spatially Varying Transforms," PCS 2009, May 6-8, 2009, Chicago, Illinois, USA.

[8] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves”, VCEG Doc. VCEG-M33, March 2001.

[9] ISO/IEC JTC1/SC29/WG11, “Joint Call for Proposals on Video Compression Technology”, MPEG Doc. N11113, Jan. 2010.

[10] W. Zheng, Y. Shishikui, M. Naemura, Y. Kanatsugu and S. Itoh, “Analysis of space-dependent characteristics of motion-compensated frame differences based on a statistical motion distribution model”, IEEE Trans. Image Processing, vol. 11, pp. 377-386, April. 2002.

Table 1 Experimental Results

Sequences	SVT	PSASVT_MID		PSASVT_AVG		PSASVT_MED		PSASVT_DIFF	
	BD-rate (%)	BD-rate (%)	Reduction of LPs tested in RDO (%)	BD-rate (%)	Reduction of LPs tested in RDO (%)	BD-rate (%)	Reduction of LPs tested in RDO (%)	BD-rate (%)	Reduction of LPs tested in RDO (%)
Traffic	-2.88	-3.06	-20.64	-3.10	-18.49	-3.04	-17.52	-2.90	-23.01
PeopleOnStreet	-1.67	-1.74	-24.35	-1.73	-21.31	-1.75	-19.82	-1.72	-27.76
Avg 2560x1600	**-2.28**	**-2.40**	**-22.50**	**-2.41**	**-19.90**	**-2.39**	**-18.67**	**-2.31**	**-25.39**
Kimono	-0.65	-0.66	-22.38	-0.70	-20.56	-0.64	-19.57	-0.67	-24.60
ParkScene	-2.22	-2.32	-20.70	-2.34	-18.78	-2.31	-17.83	-2.30	-22.99
Cactus	-2.02	-2.11	-21.58	-2.07	-19.22	-2.06	-18.19	-2.13	-24.07
BasketballDrive	-3.32	-3.59	-23.26	-3.55	-20.50	-3.51	-19.49	-3.64	-26.50
BQTerrace	-2.69	-2.86	-21.72	-2.86	-19.31	-2.81	-18.13	-2.78	-24.14
Avg 1080p	**-2.18**	**-2.31**	**-21.93**	**-2.31**	**-19.67**	**-2.27**	**-18.64**	**-2.30**	**-24.54**
Vidyo1	-2.21	-2.56	-20.57	-2.35	-18.42	-2.37	-17.49	-2.41	-23.58
Vidyo3	-5.04	-5.17	-20.90	-5.15	-18.55	-5.05	-18.33	-5.16	-23.22
Vidyo4	-3.15	-3.24	-21.39	-3.34	-18.76	-3.28	-18.81	-3.40	-24.43
Avg 720p	**-3.47**	**-3.66**	**-20.95**	**-3.61**	**-18.58**	**-3.57**	**-18.21**	**-3.66**	**-23.74**
BQMall	-3.84	-4.10	-22.10	-4.02	-19.34	-4.01	-18.24	-4.03	-25.02
PartyScene	-2.82	-2.96	-21.40	-2.94	-18.61	-2.93	-17.65	-2.84	-24.81
RaceHorses	-1.03	-1.01	-23.82	-1.01	-21.57	-1.02	-20.42	-0.96	-26.66
BasketballDrill	-2.62	-2.80	-21.05	-2.79	-19.36	-2.72	-18.28	-2.82	-22.88
Avg 832x480	**-2.58**	**-2.72**	**-22.09**	**-2.69**	**-19.72**	**-2.67**	**-18.65**	**-2.66**	**-24.84**
BlowingBubbles	-2.94	-2.97	-20.26	-2.98	-18.15	-2.98	-17.29	-2.98	-22.82
BQSquare	-2.19	-2.37	-20.52	-2.31	-18.27	-2.30	-17.20	-2.32	-22.62
RaceHorses	-1.13	-1.14	-23.64	-1.20	-21.11	-1.13	-19.90	-1.12	-26.65
BasketballPass	-2.54	-2.64	-23.19	-2.60	-20.34	-2.65	-19.20	-2.60	-26.53
Avg 416x240	**-2.20**	**-2.28**	**-21.90**	**-2.27**	**-19.47**	**-2.26**	**-18.40**	**-2.26**	**-24.66**
BigShips	-3.31	-3.72	-21.20	-3.68	-18.68	-3.59	-17.69	-3.64	-24.34
ShuttleStart	-1.71	-2.15	-20.54	-1.95	-19.70	-2.09	-6.78	-1.97	-24.08
Cyclists	-0.99	-1.24	-21.67	-1.14	-19.28	-1.28	-18.31	-1.05	-24.81
Panslow	-3.98	-4.22	-20.45	-4.18	-18.37	-4.20	-17.62	-4.08	-22.59
Sheriff	-2.46	-2.83	-21.17	-2.72	-18.99	-2.74	-18.03	-2.67	-23.78

Sailormen	-3.04	-3.43	-22.19	-3.38	-19.37	-3.38	-18.36	-3.46	-25.25
Avg 720p extra	**-2.58**	**-2.93**	**-21.20**	**-2.84**	**-19.07**	**-2.88**	**-16.13**	**-2.81**	**-24.14**
Avg all	**-2.52**	**-2.70**	**-21.70**	**-2.67**	**-19.38**	**-2.66**	**-17.92**	**-2.65**	**-24.46**

Table 2 Percentage of macroblocks coded in SVT for different sequences

Sequences	SVT		PSASVT_MID		PSASVT_AVG		PSASVT_MED		PSASVT_DIFF	
	All partitions (%)	16×16 partition (%)	All partitions (%)	16×16 partition (%)	All partitions (%)	16×16 partition (%)	All partitions (%)	16×16 partition (%)	All partitions (%)	16×16 partition (%)
Traffic	7.80	4.10	8.15	4.25	8.13	4.26	8.12	4.26	7.90	3.69
PeopleOnStreet	11.36	5.35	11.78	5.36	11.79	5.41	11.81	5.46	11.34	4.63
Avg 2560x1600	**9.58**	**4.73**	**9.97**	**4.81**	**9.96**	**4.84**	**9.97**	**4.86**	**9.62**	**4.16**
Kimono	4.43	2.30	4.40	2.01	4.42	2.04	4.45	2.09	4.21	1.73
ParkScene	10.21	5.36	10.47	5.13	10.43	5.16	10.45	5.21	10.11	4.40
Cactus	6.78	3.61	6.92	3.37	6.91	3.41	6.90	3.44	6.61	2.78
BasketballDrive	7.80	4.60	8.04	4.74	7.95	4.67	7.97	4.68	7.85	4.31
BQTerrace	7.05	4.21	7.36	4.36	7.31	4.32	7.28	4.30	7.10	3.78
Avg 1080p	**7.25**	**4.02**	**7.44**	**3.81**	**7.40**	**3.92**	**7.41**	**3.94**	**7.18**	**3.40**
Vidyo1	3.33	1.76	3.52	1.94	3.50	1.91	3.48	1.88	3.46	1.77
Vidyo3	4.08	2.40	4.36	2.71	4.31	2.66	4.28	2.61	4.43	2.66
Vidyo4	3.54	2.09	3.72	2.22	3.68	2.19	3.66	2.17	3.66	2.05
Avg 720p	**3.65**	**2.08**	**3.87**	**2.29**	**3.83**	**2.25**	**3.81**	**2.22**	**3.85**	**2.16**
BQMall	11.69	6.41	12.10	6.56	12.05	6.49	12.04	6.48	11.93	5.94
PartyScene	18.12	11.02	19.02	11.90	18.85	11.76	18.77	11.64	18.76	10.30
RaceHorses	9.48	4.32	9.53	4.10	9.60	4.13	9.57	4.13	9.26	3.60
BasketballDrill	5.51	2.48	5.68	2.38	5.66	2.38	5.66	2.39	5.53	2.14
Avg 832x480	**11.20**	**6.06**	**11.58**	**6.24**	**11.54**	**6.19**	**11.51**	**6.16**	**11.37**	**5.50**
BlowingBubbles	16.06	8.42	16.50	8.29	16.39	8.35	16.45	8.43	16.29	7.33
BQSquare	12.64	7.70	13.55	8.39	13.36	8.23	13.35	8.19	13.33	7.67
RaceHorses	12.45	5.06	12.60	4.80	12.63	4.81	12.67	4.80	12.39	4.22
BasketballPass	10.32	4.93	10.55	4.88	10.52	4.88	10.63	4.95	10.42	4.36
Avg 416x240	**12.87**	**6.53**	**13.30**	**6.59**	**13.23**	**6.57**	**13.28**	**6.59**	**13.11**	**5.90**
BigShips	11.36	6.33	11.88	6.40	11.84	6.43	11.86	6.44	11.57	5.50
ShuttleStart	3.18	1.81	3.42	2.03	3.38	2.00	3.34	1.96	3.29	1.75
Cyclists	6.57	3.66	7.21	4.29	7.08	4.17	7.03	4.10	7.00	3.94
Panslow	10.48	6.29	10.54	5.78	10.51	5.80	10.55	5.83	10.11	4.84
Sheriff	10.09	5.13	10.64	5.58	10.58	5.53	10.54	5.50	10.41	4.98
Sailormen	11.53	6.60	12.00	6.75	11.95	6.74	11.95	6.76	11.72	6.05
Avg 720p extra	**8.87**	**4.97**	**9.28**	**5.14**	**9.22**	**5.11**	**9.21**	**5.10**	**9.02**	**4.51**
Avg all	**8.99**	**4.83**	**9.33**	**4.93**	**9.28**	**4.91**	**9.28**	**4.90**	**9.11**	**4.35**