

# LOW COMPLEXITY ALGORITHM FOR SPATIALLY VARYING TRANSFORMS

Cixun Zhang<sup>\*1</sup>, Kemal Ugur<sup>§</sup>, Jani Lainema<sup>§</sup>, Antti Hallapuro<sup>§</sup>, Moncef Gabbouj<sup>\*</sup>

<sup>\*</sup>Tampere University of Technology, Tampere, Finland

<sup>§</sup>Nokia Research Center, Tampere, Finland

## ABSTRACT

In our previous work, we introduced Spatially Varying Transforms (SVT) for video coding, where the location of the transform block within the macroblock is not fixed but varying. SVT has lower decoding complexity compared to standard methods as only a portion of the prediction error needs to be decoded. However, the encoding complexity of SVT can be relatively high because of the need to perform Rate Distortion Optimization (RDO) for each candidate Location Parameter (LP). In this work, we propose a low complexity algorithm operating on macroblock and block level to reduce the encoding complexity of SVT. The proposed low complexity algorithm includes selection of available candidate LP based on motion difference and a hierarchical search algorithm. Experimental results show that the proposed low complexity algorithm can reduce around 80% of the candidate LP tested in RDO with only marginal penalty in coding efficiency.

*Index Terms*— Video coding, H.264/AVC, Spatially Varying Transform (SVT), low complexity

## 1. INTRODUCTION

High-Definition (HD) video is becoming more popular and commonly used nowadays as HD displays are widely available and available bandwidth/storage has increased rapidly. To better satisfy the requirements of increased usage of HD video especially in resource constrained applications, two key issues need to be addressed: coding efficiency and implementation complexity. In our previous work [1], [2], we proposed a novel algorithm, named as Spatially Varying Transform (SVT), which provides coding efficiency gains over H.264/AVC with lower decoding complexity. The motivations leading to design of SVT are two-fold [1]:

1. In existing video coding standards, the block based transform design does not align the underlying transform with the possible edge location. In this case, the coding efficiency of transform coding decreases.

2. Coding the entire prediction error signal may not be the best choice in terms of rate distortion tradeoff.

In [1], we utilized an 8x8 transform block with varying spatial location within the 16x16 macroblock to code the prediction error. This was shown to improve the coding efficiency of standard video coders, such as H.264/AVC, as the prediction error is localized better. In [2], we extended the concept to use variable block-size transforms within the SVT framework. It was shown that by adapting both the size of the transform block and its spatial location, the prediction error is better localized and the underlying correlations are better exploited thus the coding efficiency is improved. With SVT, as the number of transform blocks used to code the prediction error is reduced, the decoding complexity is expected to be reduced. However encoding complexity is higher mainly due to the brute force search in RDO and becomes a major concern when applying SVT to practical video codecs. In this work, we address this issue of SVT and propose low complexity algorithm operating on macroblock and block level to reduce its encoding complexity. The proposed low complexity algorithm includes selection of available Location Parameters (LP) for SVT based on motion difference and a hierarchical search algorithm. Experimental results show that encoding complexity of SVT can be greatly reduced by using the proposed low complexity algorithm while most of the coding efficiency improvement is retained.

This paper is organized as follows. A brief review of SVT is presented in section 2. Section 3 introduces proposed low complexity algorithm for SVT and presents a detail analysis. Experimental results are given in section 4 and section 5 concludes the paper.

## 2. SPATIALLY VARYING TRANSFORM

Transform coding is widely used in video coding standards to decorrelate the prediction error and achieve increased compression rates. Normally, transform coding is applied to prediction error at fixed locations. However, this is known to have several weaknesses that may hurt the coding efficiency and decrease visual quality. First of all, if the prediction error has a structure that is not suitable for the underlying transform, lots of high frequency coefficients will be generated in the transform domain and the number of bits needed to encode the coefficients grows large. In this situa-

---

<sup>1</sup> The first author would like to thank the support of Nokia Foundation Scholarship.

tion, the coding efficiency decreases. Furthermore, notorious visual artifacts such as ringing typically appear when high frequency coefficients are quantized.

In [1], we proposed SVT to reduce these drawbacks of transform coding. The basic idea of SVT is that the transform coding is not restricted to be applied at fixed locations, but instead can be applied at any location according to the characteristics of the prediction error. With this flexibility, we are able to improve coding efficiency 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 prediction error region for a sub-region and only coding this sub-region according to a certain criterion. Information of the location of the selected sub-region inside the region is coded into the bitstream if necessary.

In [1], we studied 8x8 SVT, which is illustrated in Fig. 1. As shown in the Figure, we select and code one 8x8 block inside a 16x16 macroblock with a corresponding 8x8 transform. Rate Distortion Optimization (RDO) is used to select the best LP  $(\Delta x, \Delta y)$  for SVT and determine if SVT is used for each macroblock by minimizing:

$$J = D + \lambda \cdot R \quad (1)$$

where  $J$  is the RD cost of the selected mode,  $D$  is the distortion,  $R$  is the bit rate and  $\lambda$  is the Lagrangian multiplier. It is suggested that  $(\Delta x, \Delta y)$  be selected from the set  $\Phi = \{(0..8, 0), (0..8, 8), (0, 1..7), (8, 1..7)\}^2$  which has 32 candidates in [1].

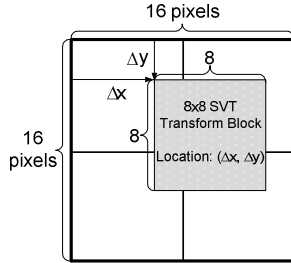


Fig.1 Illustration of 8x8 spatially varying transform

In [2], we studied 16x4 SVT and 4x16 SVT, which are illustrated in Fig. 2. As shown in the figure, we select and code one 16x4/4x16 block inside a 16x16 macroblock with a corresponding 16x4/4x16 transform. Considering both of them as a whole, the LP of the selected 16x4/4x16 block can be represented as  $(\text{shape}, \Delta y/\Delta x)$  which can be selected from the set  $\Phi = \{(0..1, 0..12)\}$  where  $\text{shape}=0$  means 16x4 block is selected and  $\text{shape}=1$  means 4x16 block is selected. There are altogether 26 LP candidates. Also, RDO is used to select the best LP  $(\text{shape}, \Delta y/\Delta x)$  for SVT and determine if SVT is used for each macroblock.

We also studied Variable Block-size Spatially Varying Transforms (VBSVT) by utilizing 8x8, 16x4 and 4x16 block sizes instead of one fixed block-size SVT and selecting the block size adaptively [2]. It was shown that by adapting both the size of the transform block and its spatial location,

the prediction error is better localized and the underlying correlations are better exploited thus coding efficiency can be improved over fixed block-size SVT.

As the number of transform blocks used to code the prediction error is reduced, the decoding complexity of SVT is expected to be reduced. However, even though we do not change the motion estimation, sub-macroblock partition decision process for the macroblocks that use SVT [1], [2], encoding complexity of SVT is higher due to the brute force search process in RDO. For example, in the case of VBSVT, there are total of 58 candidate LP for one macroblock mode and we need to conduct RDO for each candidate LP to select the best one. Note that we need to conduct transform, quantization, entropy coding, inverse transform and inverse quantization to calculate the RD cost in (1) and the complexity is high. Thus, in order to reduce the encoding complexity of SVT, the basic idea is to reduce the number of candidate LP tested in RDO.

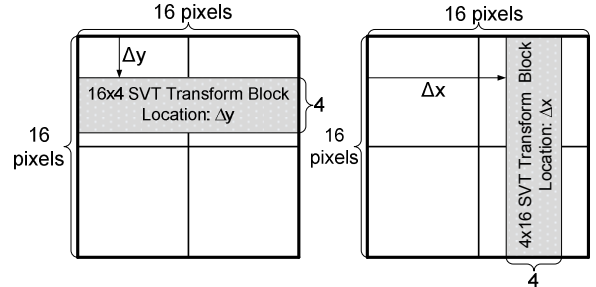


Fig.2 Illustration of 16x4 and 4x16 spatially varying transform

### 3. LOW COMPLEXITY ALGORITHM FOR SPATIALLY VARYING TRANSFORMS

Our proposed low complexity algorithm includes macroblock level algorithm and block level algorithm, which will be described in section 3.1 and 3.2 below, respectively.

#### 3.1. Macroblock level low complexity algorithm

The basic idea of our macroblock level low complexity algorithm is to skip testing SVT for macroblocks for which SVT is unlikely useful. Specifically we only try SVT for the macroblock modes, when the following two criteria are true:

$$\min(J_{inter}, J_{skip}) \leq J_{intra} \quad (2)$$

$$J_{mode} \leq \min(J_{inter}, J_{skip}) + th \quad (3)$$

where  $J_{inter}$  and  $J_{intra}$  are the minimum RD cost of available inter and intra macroblock modes in normal coding respectively and  $J_{mode}$  and  $J_{skip}$  are the RD cost of current macroblock mode to be tested with SVT and SKIP mode in normal coding respectively. The threshold  $th$  in (3) is based on Quantization Parameter (QP) and is empirically set to be

$$th = \lambda \cdot \max(23 - QP/2, 0) \quad (4)$$

In (2), by comparing inter modes, intra modes and SKIP mode, we assume that if the RD cost for intra modes is the lowest, then the probability for inter modes with SVT to

<sup>2</sup> As in [1], [2], in this paper, notation  $x..y$  is used to specify a range of integer values starting from  $x$  to  $y$  inclusive, with  $x, y$  being integer numbers.

have lower RD cost than that will be very small, so there is probably no need to check SVT for this macroblock. Similarly, in (3), we assume that if the RD cost for the current mode under test is much larger than the RD cost for the best mode, then the probability for this mode with SVT to have lower RD cost than that of best mode will also be very small.

### 3.2. Block level low complexity algorithm

The proposed block level low complexity algorithm includes two steps: selection of available candidate LP based on motion difference in the first step and a hierarchical search algorithm in the second step. These two steps are detailed in section 3.2.1 and 3.2.2 below, respectively.

#### 3.2.1. Selection of available candidate LP based on motion difference

In [1], [2], SVT was used for macroblocks with 16x16, 16x8, 8x16 and 8x8 motion compensation partitions. This was based on extensive experimental statistics that SVT is used considerably when the same macroblock partition mode is considered. One straightforward approach to reduce the encoding complexity is to restrict the transform block in a candidate SVT block to be inside the same motion compensation block boundary. Although this is simple because only macroblock partition information is used, some penalty was observed in coding efficiency especially for sequences with slow motion and rich detail, for instance, *Preakness* and *Night*. This is because prediction from different motion compensation blocks is not the major reason to cause blocking effect [4] which will probably hurt transform coding. It will even not necessarily cause blocking effect if the motion difference of different motion compensation blocks is small enough and/or the prediction is good (and in this case, we may still use SVT to improve coding efficiency). Keeping this in mind, in this work, we skip testing candidate LP when and only when the transform block(s) in the SVT block overlaps with different motion compensation blocks of which the motion difference is significant according to a certain criterion. In order to measure the motion difference we use a similar method to the one used in deriving the boundary strength parameter in deblocking filter of H.264/AVC [3], [4]. Specifically, we skip testing candidate LP and mark it unavailable if at least one of the following conditions is true:

- If the transform applied to the SVT block at that position overlaps at least two neighboring motion compensation blocks and the motion vectors of these compensation blocks are larger than or equal to a pre-defined threshold which is set to be one integer pixel in this work.
- If the transform applied to the SVT block at that position overlaps at least two neighboring motion compensation blocks and the reference frames of these two neighboring motion compensation blocks are different.

Noting that the number of available candidate LP varies from one macroblock to another and the information to derive the available candidate LP can be obtained both in encoder and decoder, the index of selected LP is coded as follows. Assume there are  $N$  ( $N > 0$ ) available candidate LP and the index of selected candidate is  $n$  ( $0 \leq n < N$ ), then it is coded as

$$\begin{cases} V = n, L = \lfloor \log_2 N \rfloor + 1, & \text{if } n < 2(N - 2^{\lfloor \log_2 N \rfloor}) \\ V = n - (N - 2^{\lfloor \log_2 N \rfloor}), L = \lfloor \log_2 N \rfloor, & \text{otherwise} \end{cases}, \quad (5)$$

where  $V$  represents the binary value of the coded bits and  $L$  represents the number of bits coded. This is based on the observation that the possibility of each available candidate LP is quite similar to each other. Adaptively entropy coding according to the statistics of available candidates might bring more gain but it is not considered here.

This approach shows stable coding efficiency for sequences with different characteristics and achieves similar gain over a wide range of test set as the original algorithm. Finally, we note that the additional complexity introduced by this approach is marginal when it is carefully implemented since: 1) the decision is simple which only uses the motion vector and reference frame information and it is only conducted when necessary; 2) generally several candidate LP representing spatially consecutive blocks can be marked available or unavailable at the same time in one decision.

#### 3.2.2. Hierarchical search algorithm

The basic idea of our hierarchical search algorithm is to first find the best LP in a relatively coarse resolution and then refine the results in a finer resolution. Specifically, we define the candidate LP in coarser resolution as “key candidate LP”, which are marked as squares in Fig. 3 and then do as follows.

1. Let  $\Phi_1$  denote the set of all available key candidate LP. Select the best one in  $\Phi_1$  with lowest RD cost and let  $\Phi_2$  denote its available neighboring candidate LP which are marked as triangles in Fig. 3.
2. The key candidate LP are divided into 14 LP zones as shown in Fig. 3. Select the best LP zone which is available and has the lowest RD cost. A LP zone is available if and only if all three key candidate LP in that zone are available and the RD cost of a LP zone is defined to be the sum of the RD cost of the three key candidate LP in that zone. Let  $\Phi_3$  denote the additional available candidate LP which are inside the best LP zone (marked as stars in Fig. 3).
3. Select the best LP which has the lowest RD cost among all the candidates in  $\Phi_1$ ,  $\Phi_2$  and  $\Phi_3$  and exit the algorithm.

More sophisticated algorithms using the same basic idea can be designed, for instance, by using different definitions of key LP and zone, and/or examining different number of good key candidate LP and zones.

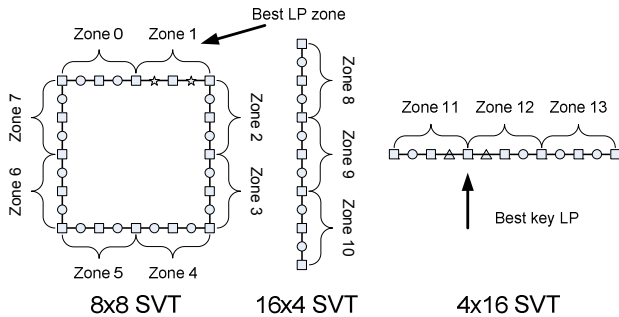


Fig.3 Illustration of hierarchical search algorithm

#### 4. EXPERIMENTAL RESULTS

We implemented the proposed low complexity algorithm for VBSVT [2] (which we denote as LC\_VBSVT) on KTA1.8 reference software [5] in order to evaluate its effectiveness. The test condition is exactly the same as that used in our previous paper [1], [2]. Important coding parameters used in our experiments are listed below:

- High Profile
- $QP_1=22, 27, 32, 37, QP_p=QP_1+1$
- CAVLC is used as the entropy coding
- Frame structure is IPPP, 4 reference frames
- Motion vector search range  $\pm 64$  pels, resolution  $\frac{1}{4}$ -pel
- RDO in the “High Complexity Mode”
- Two configurations are tested. 1) Low complexity configuration: motion estimation block sizes are  $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$ ,  $8 \times 8$ , and no  $4 \times 4$  transform is used. This represents a low complexity codec with most effective tools for HD video coding; 2) High complexity configuration: motion estimation block sizes are  $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$ ,  $8 \times 8$ ,  $8 \times 4$ ,  $4 \times 8$ ,  $4 \times 4$ , and  $4 \times 4$  transform is also used. This represents a high complexity codec with full usage of the tools provided in the standard.

We measure the average bitrate reduction ( $\Delta$ BD-RATE) of LC\_VBSVT compared to H.264/AVC according to [6]. We also measure the average reduction of candidate LP tested in RDO over all test QP using LC\_VBSVT. The results are shown in Table 1 and Table 2 respectively. We can see that with the proposed low complexity algorithm we can reduce about 80% of the candidate LP tested in RDO while retaining most of the coding efficiency, for both low complexity configuration and high complexity configuration. The reduction in high complexity configuration is slightly less than that in low complexity configuration because when  $4 \times 4$  transform is used, more candidate LP is possible to be available according to our selection method based on motion difference described in section 3.2.1 and still need to be tested in RDO.

#### 5. CONCLUSIONS

In this paper, we propose low complexity algorithm for Spatially Varying Transforms (SVT) and it is shown to reduce around 80% of the candidate Location Parameters (LP)

tested in RDO with only marginal penalty in coding efficiency. Compared to H.264/AVC, the low complexity SVT achieves on average 3.69% and 2.34% bit-rate reduction for low complexity configuration and high complexity configuration, respectively.

#### 6. REFERENCES

- [1] C. Zhang, K. Ugur, J. Lainema and M. Gabbouj, “Video coding using spatially varying transform”, PSIVT 2009.
- [2] C. Zhang, K. Ugur, J. Lainema and M. Gabbouj, “Video coding using variable block-size spatially varying transform”, ICASSP 2009, to appear.
- [3] “Advanced video coding for generic audiovisual services”, ITU-T Recommendation H.264, Mar. 2005.
- [4] P. List, A. Joch, J. Lainema, G. Bjontegaard and M. Karczewicz, “Adaptive Deblocking filter”, IEEE Trans. Circuits Syst. Video Technol. Vol. 13, no. 7, pp. 614-619, Jul. 2003.
- [5] KTA reference model 1.8 [online], available at <http://iphome.hhi.de/suehring/tml/download/KTA/>
- [6] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves”, VCEG Doc. VCEG-M33, March 2001.

Table 1 Experimental Results (Low complexity configuration)

Sequence	VBSVT [2]	LC_VBSVT	Reduction
BigShips	-4.96%	-4.75%	82.8%
ShuttleStart	-3.42%	-3.25%	83.0%
City	-4.45%	-4.14%	82.6%
Night	-4.12%	-3.53%	84.5%
Optis	-3.97%	-3.50%	82.6%
Spincalendar	-3.49%	-3.16%	84.4%
Cyclists	-2.52%	-2.48%	83.3%
Preakness	-3.71%	-3.07%	80.6%
Panslow	-6.19%	-5.38%	85.3%
Sheriff	-3.43%	-3.09%	83.1%
Sailormen	-4.94%	-4.26%	84.5%
Average	-4.11%	-3.69%	83.4%

Table 2 Experimental Results (High complexity configuration)

Sequence	VBSVT [2]	LC_VBSVT	Reduction
BigShips	-3.04%	-2.99%	80.1%
ShuttleStart	-1.46%	-1.98%	82.2%
City	-2.37%	-2.07%	80.0%
Night	-2.59%	-2.39%	82.4%
Optis	-3.04%	-2.62%	80.1%
Spincalendar	-2.24%	-2.00%	81.9%
Cyclists	-1.34%	-1.36%	81.2%
Preakness	-1.78%	-1.54%	78.7%
Panslow	-3.16%	-3.25%	82.0%
Sheriff	-2.26%	-2.16%	81.1%
Sailormen	-3.48%	-3.41%	82.5%
Average	-2.43%	-2.34%	81.1%