|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  7th Meeting: Geneva, CH, 21-30 Nov, 2011 | Document: JCTVC-G598 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Intra prediction based on weighted template matching predictors (WTM)** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal, | | |
| *Author(s) or Contact(s):* | T. Guionnet (INRIA)  L. Guillo (IRISA/CNRS)  INRIA Rennes Bretagne Atlantique  Campus de Beaulieu  35042 Rennes Cedex  France | Tel: Email: | (+33) (0) 299847100 Thomas.Guionnet@inria.fr |
| *Source:* | INRIA | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This contribution presents an intra prediction method based on weighted template matching predictors (WTM) and the results once integrated in HM4.0. The resulting codec always improves the intra prediction. Without the class F, the average BD-rate gain is 0.5% for High Efficiency (HE) and 0.6% for Low Complexity (LC) profiles. With the class F, the average BD-rate gains are 1% for HE and 1.1% LC profiles. BD-rate gains for the only class F are 3.1% and 3.4% for HE and LC profiles respectively.

Preliminary tests were run when SDIP is jointly activated with WTM in HM4.0. BD-rate gains are largely cumulative.

# Introduction

Weighted template matching (WTM) is based on a linear combination of three predictors obtained thanks to template matching. It is analogous to the method presented during the 4th meeting of JCT-VC [1]. However, section 2 details differences between previous template matching approaches and WTM. This new version is implemented in HM4.0 and section 3 gathers experimental results for all classes, with and without the class F dedicated to screen content, with the intra-only configuration as defined in [4]. BD rate gains [6] are provided and also statistics about the number of blocks predicted with WTM. Section 4 presents preliminary tests when WTM and SDIP are both activated in HM4.0. Section 5 outlines the main advantages of this approach and concludes this document.

# Intra prediction based on weighted template matching predictors (WTM)

## Rationale of WTM

In template matching approaches, intra prediction process is a 3 step method. First an area surrounding the block to be predicted, B, is defined. This is the template Y, which is often an L-shaped block (represented in Figure 1). The second step aims at finding in the causal neighbourhood, i.e. among blocks which have already been reconstructed (Figure 1), pieces of picture that best matches the template. These pieces of picture are called below candidate templates Ai. Matching criteria can use metrics such as the SAD [1] or MSE [3] between pixels of the template Y and the candidate template Ai. Then, blocks Bi surrounded by the best matching areas are used to compute predictors Pi, which might be averaged as described in [1] to get the prediction P of the block B. Intra prediction based on template matching relies on the assumption that correctly approximating the template leads to a good prediction.

Block to be predicted B

Template Y

*Causal neighbourhood*

Candidate Template A1

Block B1

Candidate Template AN

Block BN

Figure 1: Search regions from causal neighbourhood

Although WTM is analogous to this general approach, it differs from it for the shapes of templates, the matching criterion and how predictors are obtained thanks to a weighting factor and then how they are combined to get a predictor as a linear combination. These differences are detailed in the following sections.

## Four shapes of templates

Most of templates are usually L-shaped. However that shape assumes that pixels surrounding the block are as important as each other for prediction whatever the directional characteristics of the block to be predicted. That might be the case. That is why WTM still uses an L-shaped template which is one pixel large. However, three other templates are added. The width and the height of respectively the right and upper parts of the L shape can be extended to 4 pixels. Hence, the extended parts contribute more to the prediction process, which, that way, takes more into account possible directional characteristics of the block to be predicted. Widths or heights of sub parts of templates are the same, whatever the size of the PU used by WTM (4x4, 8x8, 16x16 and 32x32). The Figure 2 shows the four shapes of template used by WTM.

UDL

L

U

UL

Figure 2 : Shape of templates

These shapes will be separately used in competing predictions. A WTM predictor is always made of a combination of three predictors obtained using the same template shape.

## Selection of templates

As for any template based solution, a search area is defined in the causal neighborhood of the block B. Ideally, all the past processed image should be considered in order to maximize coding efficiency. However, in the HEVC context, arbitrary sizes have been fixed in order to limit processing complexity and memory consumption, while retaining sufficient coding efficiency.

In order to make the implementation easier, the search area is divided into 2 or 3 rectangular and contiguous search windows. The characteristics of the search areas are summarized in Table 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Block B size** | **Search windows number** | **Search windows width** | **Search windows height** |
| 4x4 | 3 | 12 | 4 |
| 8x8 | 2 | 20 | 8 |
| 16x16 | 2 | 8 | 16 |
| 32x32 | 2 | 4 | 32 |

Table 1: Search areas characteristics

The search windows are arranged relatively to the block B, immediately above or on the left, as illustrated on . Note that this figure depicts the area scanned by the lower-right corner of the templates.

Block B

Window 1

Window 2

Window 3

(If available)

Figure 3: Search windows positions relatively to block B.

So the principle is to scan the search area and to retain the N best candidate blocks Bi (here with N = 3). The best candidate blocks Bi are those with the lowest Sum of square differences (SSD) between their associated template Ai and the block to be predicted template Y.

The windows have been chosen small enough to keep the complexity reasonable. Moreover, in order to further reduce complexity, the search process is divided into two passes:

* In a first pass, each window is scanned on a quincunx grid starting from its upper-left corner. Therefore, the search space is subsampled by a factor 2. At the end of this pass, N candidate blocks Bi are selected.
* Secondly, for each selected block Bi, the search is refined by visiting the four neighbouring positions of Bi which were not scanned during the first pass.

This strategy has proven to provide almost the same performance as the classical full search, while being significantly faster.

Note finally that the N best candidate blocks are searched for each of the four available template shapes. As these shapes are overlapping, it is straightforward to factorize the four searches in a single pass.

## Prediction as a linear combination of template matching predictors

Given 3 candidate blocks Bi, 3 predictors Pi are computed and then linearly combined as a single predictor P of the block B. Let us consider, for clarity, that the templates and blocks are represented in a vector form.

First, each block Bi is scaled by a factor in order to obtain a predictor Pi of the block B. The factor is computed using the templates and the least mean squares (LMS) criterion. We have:

And the predictor Pi is given by

Finally, we want to compute a predictor of B as:

The weight wi is simply set to , as the gain brought by more complex methods has not proven to be significant enough in the HEVC context.

Averaging several templates has proven to increase significantly the coding efficiency of template matching. However, in some cases, one may have a part of the N predictors which are so different from the block B that they impair the quality of the prediction. A simple thresholding strategy has been added in order to reject such outliers. The best candidate (in the template SSD sense) is always kept. The other candidates are compared to the best candidate. Those too far from it are rejected.

Finally, although the above description was done using formal equations, one has to note that these computations can be implemented easily using integer arithmetic, thanks to appropriate scaling and rounding, without loss of performance.

## Syntax modification

WTM predicts a block as a linear combination of template matching predictors Pi. Four shapes of template are available. However, the templates related to these predictors Pi have all the same shape. Consequently, two pieces of information must be signaled to the decoder: when a block is predicted with WTM and which shape of template was used.

To do so, four direction modes have been overloaded: from the mode 10 up to the mode 13. They are associated to a shape (cf. Figure 2) as listed in the table below:

|  |  |
| --- | --- |
| **Intra mode** | **Shape** |
| 10 | UDL |
| 11 | U |
| 12 | L |
| 13 | UL |

Table 2 : Relation between intra mode and shape of template

An extra bit is added for all of these four modes and set to true if WTM is used as described below:

|  |
| --- |
| prediction\_unit( x0, y0, log2PUWidth, log2PUHeight, PartIdx ,    InferredMergeFlag ) { |
| if( skip\_flag[ x0 ][ y0 ] ) { |
| if(NumMPMCand > 1 ) |
| **merge\_idx[** x0 ][ y0 **]** |
| } else if( PredMode = = MODE\_INTRA ) { |
| **prev\_intra\_luma\_pred\_flag**[ x0 **][** y0 ] |
| if( prev\_intra\_luma\_pred\_flag[ x0 ][ y0 ] ) |
| if( NumMPMCand > 1 ) |
| **mpm\_idx[** x0 **][** y0 **]** |
| else |
| **rem\_intra\_luma\_pred\_mode**[ x0 **][**y0 ] |
| if(PredMode is associated to WTM) |
| **WTM\_mode\_flag[x0][y0]** |
| } else { /\* MODE\_INTER \*/ |
| ,,, |
| } |

Table 3: Signaling taking into account WTM

# Experimental results

This section presents results of the WTM integration in the HM4.0. The “All Intra – High efficiency” (AI-HE) and “All Intra – Low complexity” (AI-LC) as defined in [1] were used.

WTM is implemented for the following sizes of prediction unit (PU): 4x4, 8x8, 16x16 and 32x32. The table below shows when WTM is activated (notified with a “√”) according to the video class and the PU size:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **HE** | | | | | **LC** | | | | |
|  | **4x4** | **8x8** | **16x16** | **32x32** | **64x64** | **4x4** | **8x8** | **16x16** | **32x32** | **64x64** |
| **Class A** | - | √ | √ | √ | - | - | √ | √ | - | - |
| **Class B** | √ | √ | √ | √ | - | √ | √ | √ | - | - |
| **Class C** | √ | √ | √ | - | - | √ | √ | - | - | - |
| **Class D** | √ | √ | √ | - | - | √ | √ | - | - | - |
| **Class E** | √ | √ | √ | √ | - | √ | √ | √ | - | - |
| **Class F** | √ | √ | √ | √ | - | √ | √ | √ | - | - |

Table 4 : Activation of WTM according to video classes and PU sizes

The Results with only the classes A, B, C, D and E are presented in the Table 5, while the results including class F are reported in Table 6.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0,0% | -2,0% | -2,6% | -0,1% | -1,4% | -2,0% |
| Class B | -0,8% | -0,9% | -0,8% | -0,9% | -0,7% | -0,6% |
| Class C | -0,9% | -0,5% | -0,7% | -1,0% | -0,6% | -0,8% |
| Class D | -0,4% | -0,3% | -0,4% | -0,5% | -0,4% | -0,4% |
| Class E | -0,5% | -0,4% | -0,4% | -0,7% | -0,6% | -0,2% |
| **Overall** | -0,5% | -0,8% | -1,0% | -0,7% | -0,7% | -0,8% |
|  | -0,5% | -0,8% | -1,0% | -0,7% | -0,7% | -0,8% |
| Enc Time[%] | 114% | | | 121% | | |
| Dec Time[%] | 104% | | | 105% | | |

Table 5 : HM4.0+WTM vs HM4.0 all classes except class F

.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0,0% | -2,0% | -2,6% | -0,1% | -1,4% | -2,0% |
| Class B | -0,8% | -0,9% | -0,8% | -0,9% | -0,7% | -0,6% |
| Class C | -0,9% | -0,5% | -0,7% | -1,0% | -0,6% | -0,8% |
| Class D | -0,4% | -0,3% | -0,4% | -0,5% | -0,4% | -0,4% |
| Class E | -0,5% | -0,4% | -0,4% | -0,7% | -0,6% | -0,2% |
| Class F | -3,1% | -2,9% | -2,8% | -3,5% | -3,5% | -3,4% |
| **Overall** | -1,0% | -1,2% | -1,3% | -1,1% | -1,2% | -1,2% |
|  | -1,0% | -1,2% | -1,3% | -1,1% | -1,2% | -1,2% |
| Enc Time[%] | 115% | | | 121% | | |
| Dec Time[%] | 104% | | | 106% | | |

Table 6 : HM4.0+WTM vs HM4.0 all classes (class F included)

These results show that:

* WTM always improved the intra prediction of HM4.0,
* The average BD rate gains are 0.5% in HE and 0.6% in LC profiles when all classes except class F are encoded,
* The average BD rate gains are 1.0% in HE and 1.1% in LC profiles when all classes are encoded (class F included).

An analysis of the results per video (see attached excel files) shows that:

* When only classes A, B, C, D and E are encoded, the best BD rate gains are reached for the video “basketballdrill” and are 2.4% in HE and 2.3% in LC profiles.
* When all classes are encoded (class F included), the best BD rate gains are reached for the video “SlideEditing” and are 5.8% in HE and 6.5% in LC profiles.

Statistics about the number of blocks predicted thanks to WTM were recorded. They give first the global percentage and how it is dispatched according to the different shapes of template. Figures are gathered in Table 7 and Table 8, with or without class F respectively.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **All shapes** | **UDL** | **U** | **L** | **UL** |
| **AI HE** | 3.89% | 1.05% | 1.00% | 0.93% | 0.91% |
| **AI LC** | 2.89% | 0.88% | 0.79% | 0.64% | 0.58% |

Table 7 : Percentage of blocks predicted with WTM (except class F)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **All shapes** | **UDL** | **U** | **L** | **UL** |
| **AI HE** | 4.47% | 1.21% | 1.12% | 1.11% | 1.03% |
| **AI LC** | 3,34% | 0,99% | 0,91% | 0,76% | 0.68% |

Table 8 : Percentage of blocks predicted with WTM (all classes, F included)

Differences are related to the class F. The same figures for the class F only are provided in Table 9:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **All shapes** | **UDL** | **U** | **L** | **UL** |
| **AI HE** | 7.38% | 2.02% | 1.71% | 2.02% | 1.63% |
| **AI LC** | 5.55% | 1.57% | 1.49% | 1.34% | 1.16% |

Table 9 : Percentage of blocks predicted with WTM (class F only)

WTM is more often used to predict blocks for the class F than for the other classes. Percentage of blocks predicted with WTM can reach up to 13.78% for the video “SlideEditing” encoded at QP 22 in HE profile.

Characteristics of video belonging to the class F are better modeled by WTM than current intra prediction method implemented in the current HM4.0. These videos include non-natural content such as scenes from video games, characters and ideograms. They may contain real scenes, such as “BasketBallDrillText” but texts or logos have been superimposed. WTM always improved the intra prediction of HM4.0 whatever the video class. Nonetheless, WTM is more relevant with videos which include non-natural content.

# Preliminary tests with SDIP

SDIP integrated in HM4.0 was evaluated in the core experiment 6 (CE6) [5]. A very first integration of WTM in that version leads to the following results:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | |
|  | Y | U | V | Y | U | V |
| Class A | -0.6% | -1.5% | -2.0% | -1.3% | -1.9% | -2.5% |
| Class B | -1.9% | -2.9% | -3.1% | -2.7% | -2.6% | -3.1% |
| Class C | -2.3% | -1.7% | -2.2% | -3.0% | -2.7% | -3.2% |
| Class D | -1.8% | -1.4% | -1.3% | -2.4% | -2.2% | -2.2% |
| Class E | -2.7% | -5.3% | -5.0% | -4.2% | -6.9% | -5.7% |
| Class F | -8.1% | -6.6% | -6.9% | -8.5% | -6.8% | -7.0% |
| **Overall** | -2.9% | -3.1% | -3.3% | -3.6% | -3.7% | -3.9% |
|  | -2.9% | -3.1% | -3.3% | -3.6% | -3.7% | -3.8% |
| Enc Time[%] | 141% | | | 160% | | |
| Dec Time[%] | 105% | | | 108% | | |

Table 10: HM4.0+SDIP+WTM vs HM4.0

Gains of WTM are largely added to those provided by SDIP. When SDIP and WTM are activated, the average BD rate gains are respectively 2.9% in HE and 3.6% in LC profiles when all classes are encoded. BD rate gains can reach 15.3% in luma for the video “SlideEditing” in HE and 15.9% in LC profiles. A further analysis of the integration of WTM and SDIP in the HM would require more tests and probably a further study about how they might interact together.

# Conclusion

WTM is an intra prediction method based on weighted template matching predictors for which four different shapes of templates can be used. WTM improves intra prediction for all video classes. The average BD rate gains are 1% in HE and 1.1% LC profiles when all classes are used and 0.5% in HE and 0.6% in LC profiles also when all classes are tested except the class F. WTM performs better when videos contain non-natural content such as text, logos and scenes from video games. In these cases, average BD-rate gains are 3.1% and 3.4% in HE and LC profiles respectively.

A very first test was run when WTM and SDIP are both activated in HM4.0. Gains of WTM are largely added to those provided by SDIP whatever the video classes. However, performances related to the class F are particularly improved.

The gains brought by WTM in intra prediction and particularly for screen content videos show that further studies as part of a CE would be relevant.

# References

1. L. Guillo, R. Boitard, T. Poirier, “ Integration into the TMuC of an Intra Prediction based on a linear combination of Template Matching predictors”, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-D193, 4th Meeting: Daegu, KR, Jan, 2011.
2. T.K. Tan, C.S. Boon and Y. Suzuki, “Intra prediction by averaged template matching predictors”, CCNC, 2007.
3. C. Lan, J. Xu, F. Wu and G. Shi, “Intra frame coding with template matching prediction and adaptive transform”, ICIP 2010.
4. F. Bossen, “Common conditions and software reference configurations”, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-C900, 6th Meeting: Torino, IT, July., 2011.
5. A. Tabatabai, E. François, K. Chono, H. Yu, J. Rajan, J. Lainema, “CE6: Intra Coding Improvements”, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-C900, 6th Meeting: Torino, IT, July., 2011.
6. G. Bjontegaard, “Calculation of average PSNR differences between RD-curves”, ITU-T VCEG, 2001.

# Patent rights declaration(s)

**INRIA may have IPR relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**