Joint Collaborative Team on Video Coding (JCT-VC)
J G 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11
e g: Geneva, CH, 27 April – 7 May 2012

Document: JCTVC-I0546

Title: **Polygon Texture Unit Visual Encoding**

Status: Input Document to JCT-VC

Purpose: Proposal

Author(s) or Contact(s): Sung-Hoon Hong, Byung Kwan Yi, Panos Nasiopoulos

Tel: 1-858-354-9532
Email: hong@intribest.com, bkyi@ucsd.edu
panos@ece.ubc.ca

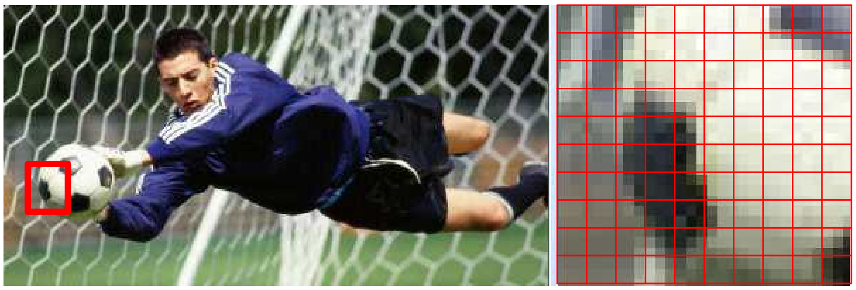
Source: Centri Alliance, Inc.

Abstract

This contribution presents a new video compression approach, which is based on a triangle (or arbitrary) shape polygon texture unit for reducing spatial and temporal redundancies. The proposed method partitions raw video frame data to polygon texture blocks first, and then predicts, transforms, quantizes and compresses the partitioned polygon textures. Performance evaluations have shown that the proposed method achieves approximately 20% bit-savings (on average) compared to HEVC.

1 Introduction / Problem Statement / …

Existing video compression methods use the square block unit approach to reduce spatial and temporal redundancies. This method has been shown to work well, although in many cases a cascade of different size blocks is applied in order to improve prediction accuracy. In that respect, a trade-off of complexity, overhead and accuracy had been reached, limiting the ability of the present efforts to surpass this upper bound and in turn have the chance to significantly improve the compression efficiency beyond what has been achieved at this point. The proposed method addresses this issue by using a polygon-based texture unit, replacing the square/parallelogram blocks with triangle-shape blocks (see Fig. 1). Its smart implementation seems to be a better way to trace similarities within a frame and between frames, resulting in a much more efficient overall compression. The proposed method includes a triangle polygon texture unit of partitioning, inter/intra prediction, DCT (discrete cosine transform) / DST (discrete sine transform), quantization and entropy coding. When the proposed algorithmic tool is used, the proposed method achieves approximately 20% bit-saving on average compared to HEVC.



(a)

(b)

Figure 1. (a) Existing square block unit visual encoding (b) The triangle polygon texture unit visual encoding

2 Our Proposed Method

In this proposal, the triangle polygon unit is defined as the basic processing unit for calculating spatial and temporal redundancies, transformation to the frequency domain, quantization and entropy coding. Each such unit may have variable pixel size. The details of our proposed encoding framework are described in the following subsections.

3 Intra-frame coding

Problem statement

Each tile of the seed (I) frame, $\{I(x,y,k)\}_{x=1,2,\dots,w,\ y=1,2,\dots,h}$, of the *k*-th frame is divided into trianglular textures (polygons) . as

$$I(x,y,k) = \sum_{i=1}^{P} I^{(i)}(x,y,k) \quad (1)$$

where $I^{(i)}(x,y,k)$ is the *i*-th polygon in the *k*-th frame, and *h* and *w* are height and width of the seed frame respectively. The polygon $I^{(i)}(x,y,k)$ is represented by

$$I^{(i)}(x,y,k) = w^{(i)}(x,y)I(x,y,k) \quad (2)$$

where $w^{(i)}(x,y)$ is the *i*-th polygon of triangle mask to be represented as $\Delta(y^{(i)}, x^{(i)})$ where $y^{(i)} = [\,0\ 0\ 0\,]$ and $x^{(i)} = [\,0\ 0\ 0\,]$ are the three corner positions of the triangle polygon and uniquely defines the *i*-th polygon. Given the boundary lines, $I_A^{(i)}$, $I_B^{(i)}$, and $I_C^{(i)}$ (if the polygons including the boundary lines are available), the $I^{(i)}(x,y,k)$ inside the boundary lines are predicted to get

$\widehat{I^{(i)}}(x,y,k)=I_A^{(i)}(x_A,y_A,k)$ if $I_A^{(i)}$ is encoded or $I_B^{(i)}(x_B,y_B,k)$ if $I_B^{(i)}$ is encoded or $I_C^{(i)}(x_C,y_C,k)$ if $I_C^{(i)}$ is encoded. $\{x_A,y_A\}$ are selected as same as HEVC's intra-prediction with the multiple-angles. If more than one boundaries are available, the mode to select $I_A^{(i)}$ or $I_B^{(i)}$ or $I_C^{(i)}$ are added. The intra-prediction error in the selected tile, e, is formulated as

$$\begin{aligned}\vec{e}&=\sum_{i=1}^{P}e^{(i)}(x,y,k)\\&=\sum_{i=1}^{P}\left\{I^{(i)}(x,y,k)-\widehat{I^{(i)}}(x,y,k)\right\}\\&=\sum_{i=1}^{P}\left\{w^{(i)}I(x,y,k)-\widehat{I^{(i)}}(x,y,k)\right\}\end{aligned} \quad (3)$$

where $w^{(i)}(x,y)$ is the *i*-th polygon of triangle mask to be represented as $\Delta(y^{(i)},x^{(i)})$. And so the problem for the intra-frame coding would be to find *P*, the number of polygons in the selected tile, $y^{(i)}=[0\,0\,0]$ and $x^{(i)}=[0\,0\,0]$, the three corner points of the triangle polygon, the intra-prediction modes including the boundary selection mode and the prediction angle. In order to solve the proposed problem statement, the realistic solution is proposed.

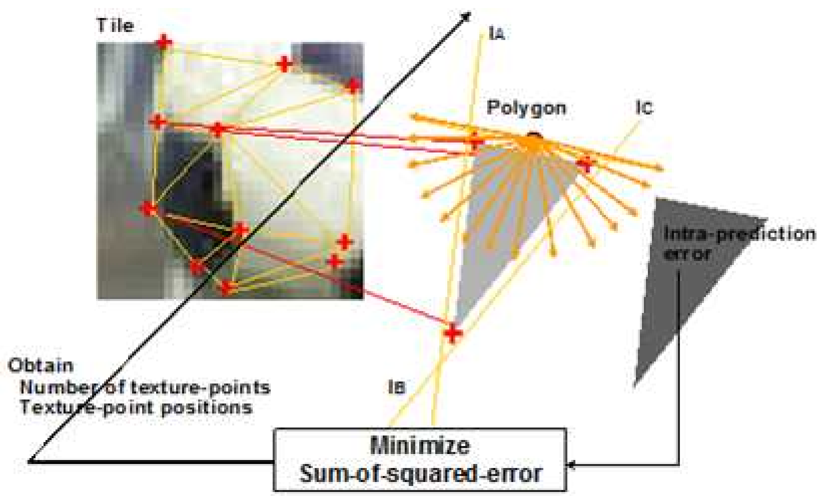


Figure 2 Overall diagram of the polygon unit intre-frame coding problem statement.

Partitioning of the I (seed) frame into triangles

Each tile of the seed (I) frame is divided into triangular textures (polygons). For each such frame, the horizontal-direction differentiation value (Gx) and the vertical-direction (Gy) of all the pixels, are calculated as $G_x(x,y,k)=I(x,y,k)-I(x-1,y,k)$ and $G_y(x,y,k)=I(x,y,k)-I(x,y-1,k)$. The pixel positions for the differentiation values greater than a threshold value (it could be adjusted for coding) are selected as the texture-points, $T_n(x,y,k)=1$ if $G_x(x,y,k)>\theta_x$ and $G_y(x,y,k)>\theta_y$. For the selected texture-points, three texture points are initially detected to construct the first triangle polygon. Another texture point is estimated to construct an additional triangle through a search process (among candidates within a range) that yields the minimum sum-of-squared-error between the intra-predicted polygon and the original one. This polygon construction process is repeated until the entire frame is fully covered by the triangle polygons.

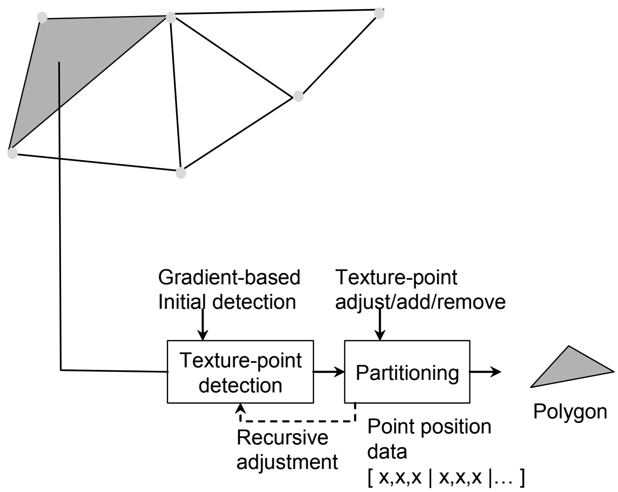


Figure 3. Polygon texture partitioning procedure.

Mode selection (similar to HEVC)

Our method uses the same modes and prediction process as HEVC. As shown in Fig. 4, in the triangle polygon structure, the boundaries of pixels (from one to three) are available if the adjacent triangles had already been encoded.

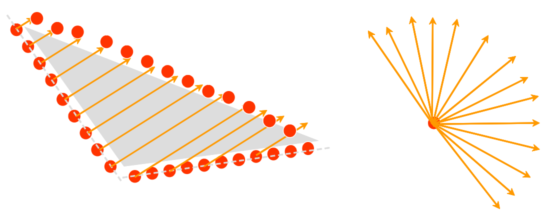


Figure 4. Polygon texture unit intra prediction: multiple angle directions for the intra prediction.

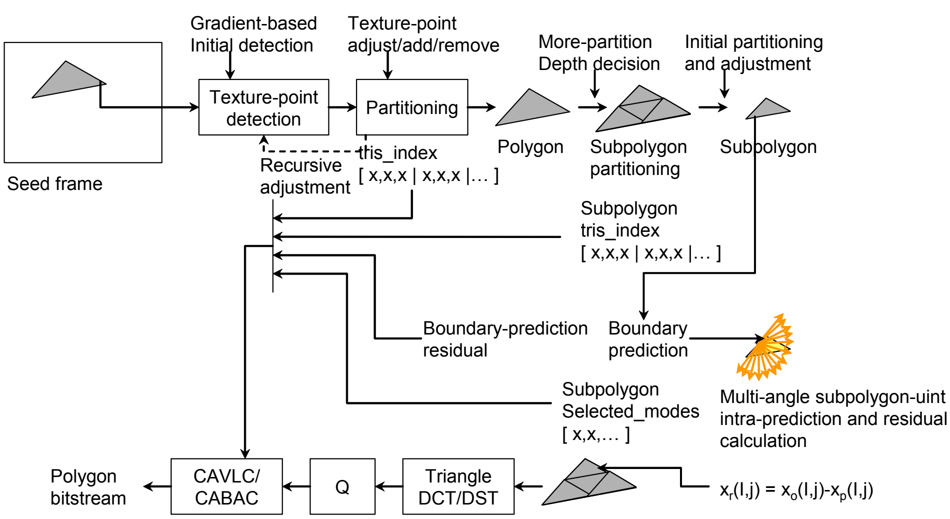


Figure 5. Overall description of the seed frame encoding of the polygon texture unit visual encoding.

Fig. 5 illustrates the proposed seed frame encoding process. Frame pixels are first partitioned by the arbitrary shape of polygon textures and then are repartitioned in triangle shape of polygons. The abrupt changes of the gradients in vertical and horizontal directions are initially detected as texture points. These detected points are adjusted and added with the objective to minimize the intra prediction residual. The partitioned polygon textures are then repartitioned and adjusted to smaller sub-polygons to further reduce the intra prediction residual. The boundaries of the texture polygons are predicted first from the boundaries of the adjacent texture polygons, and then the pixels inside the texture polygon are predicted from the boundaries of the texture polygon.

Fig. 6 illustrates the above explained polygon texture based intra prediction selection mode process. The pixel values inside the polygon texture are predicted by the boundary values of the polygon with the various angle directions.

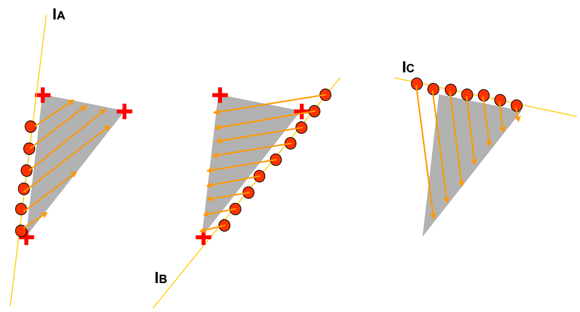


Figure 6. Polygon texture unit intra prediction.

The texture polygon prediction residual is then transformed (DCT), quantized and bit compressed. The *i*-th polygon signal in the *k*-th frame, $e^{(i)}(x,y,k)$, is then shifted to the most left position and grouped to row vectors. Then the row vectors are transformed in the horizontal-direction while using a 1-dimensional DCT of length N resulting in corresponding vector with N transform coefficients each. Next the same procedure is repeated in the vertical-direction. Those N_m elements of the row vectors which belong to the same column *m* are shifted to the most upper position and formed to column vectors. Finally the column vectors are again transformed by a 1-dimensional DCT, but now in the vertical direction and with a transform length of N_m yielding the column vectors with the entire transform coefficients of triangle DCT. The inverse triangle DCT is obtained by performing the inverse operations of the four above processing steps in reverse order. The DCT transformed prediction residuals are zigzag scanned to obtain the 1-dimensional signal, and the scanned signals are quantized with the same method as HEVC. The variable size quantized DCT transformed prediction residuals are divided into fixed size groups and then bit compressed same HEVC.

The texture points positions data to construct the texture polygon are bit compressed. The texture-point position data (relative coordinates dx_i and dy_i) are calculated as $dx_i=x_i-x_{i-1}$ and $dy_i=y_i=y_{i-1}$. The resulting residual values are vector quantized and compressed.

4 Inter Prediction

Problem Statement

As in the case of intra-frame coding, each tile of the seed (I) frame, $\{I(x,y,k)\}_{x=1,2,\dots,w,\, y=1,2,\dots,h}$, of the *k*-th frame is divided into triangular textures (polygons) as

$$I(x,y,k)=\sum_{i=1}^{P} I^{(i)}(x,y,k) \quad (4)$$

where $I^{(i)}(x,y,k)$ is the *i*-th polygon in the *k*-th frame, and *h* and *w* are height and width of the seed frame respectively. The polygon $I^{(i)}(x,y,k)$ is represented by

$$I^{(i)}(x,y,k)=w^{(i)}(x,y)I(x,y,k) \quad (5)$$

where $w^{(i)}(x,y)$ is the *i*-th polygon of triangle mask to be represented as $\Delta(y^{(i)},x^{(i)})$ where $y^{(i)}=[\,0\,0\,0\,]$ and $x^{(i)}=[\,0\,0\,0\,]$ are the three corner positions of the triangle polygon and uniquely defines the *i*-th polygon. Given the $I^{(i)}(x,y,k)$, the inter-predicted signal for the *i*-th polygon is defined as

$$\widehat{I^{(i)}}(x,y,k)=I(x_t,y_t,n) \quad (6)$$

where *n* is the reference frame number. In order to estimate $\{x_t,y_t\}$ for whole set of polygons the three corner positions of $\{x_t,y_t\}$ are initially defined first and then the other sets of $\{x_t,y_t\}$ are affinely transform using matrix *M*. The affine transform matrix *M* is calculated to solve $[\vec{x_t},\vec{y_t},1]^t = M[\vec{x},\vec{y},1]^t$ where x_t is the *x*-direction positions of three corner points of the polygon and y_t is the *y*-direction positions of three corner points of the polygon in the reference frame. $\{x_t,y_t\}$ are adjusted in search range to find the best set that minimizes the inter-prediction error. The block diagram and different steps of this process are shown in Fig. 7.

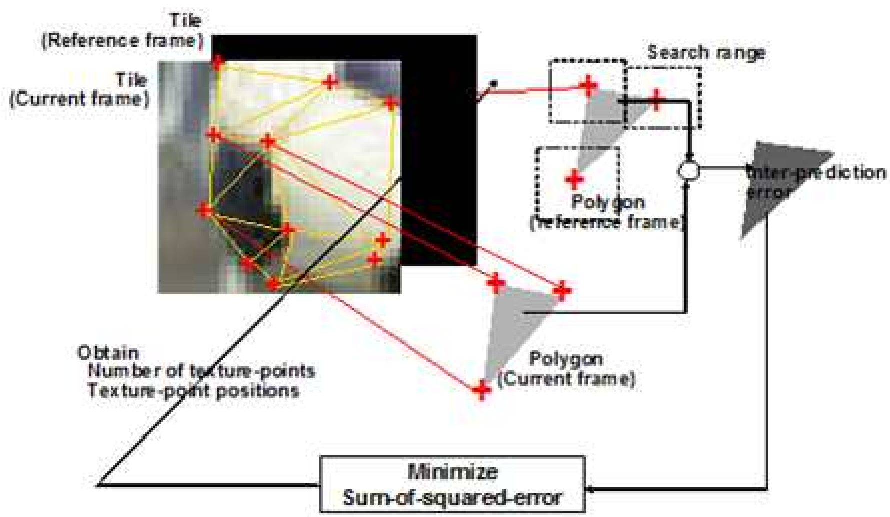


Figure 7. Overall diagram of the polygon unit inter-frame coding problem statement.

Motion Estimation

Inter-prediction is used to remove temporal redundancies The current frame is divided into polygon pixel blocks and the polygon matching is performed independently for each polygon. Polygon block matching is done by identifying a polygon pixel block from the reference frame that best matches the current polygon block. This matching is achieved by estimating texture-points movement vectors as shown in Fig. 8. The reference polygon pixel block is generated by an affine transform from the current polygon block's location in the reference fame. The affine transform parameter or the texture-points movement vector is provided by the polygon motion vector and consists of three pairs (x,y) of horizontal and vertical displacement values.

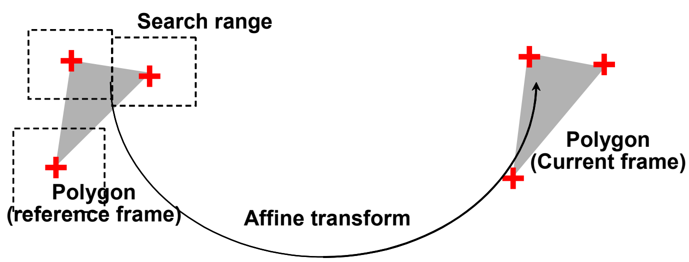


Figure 8. Triangle polygon unit inter-prediction.

If the closest match has sufficiently low distortion, the given polygon by the polygon in the initial frame is described (say, its three corner points are described). If the closest match has large distortion, then the given polygon using sum of the reference frame and the residual signal are described. If too many polygons are described separately rather than from the initial frame, then the given frame is declared a new initial frame and we repeat from subsequent frame encoding step.The proposed method uses the same GOP structure as HEVC. Fig. 8 illustrates the block diagram of our matching process.

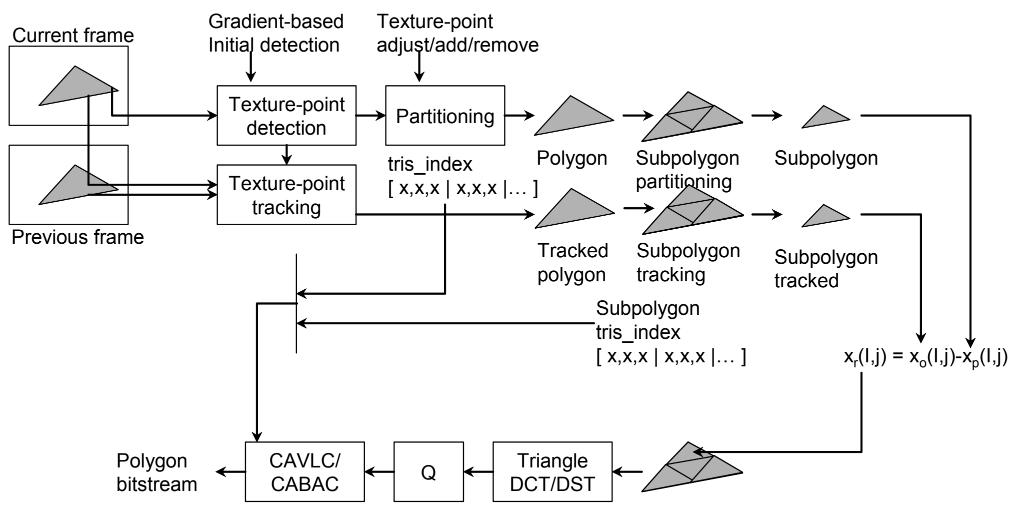


Figure 9. Block diagram of temporal matching.

Fig. 10 illustrates the inter prediction of the partitioned polygon texture. Affine transform is used to fit the polygon texture pixel values in the reference frame to the ones in the current frame.

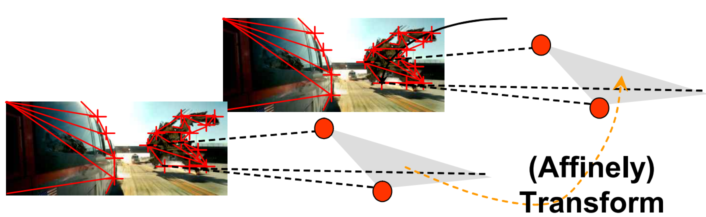


Figure 10. Inter prediction of the partitioned polygon texture.

Fig.11 illustrates how the polygon texture point detection identifies abrupt changes of gradients of pixels in the spatio-temporal domain.

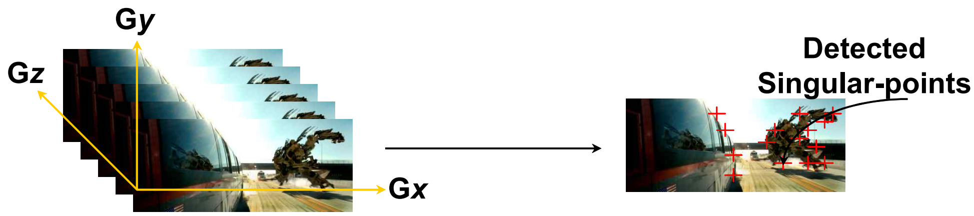


Figure 11. Polygon texture point detection.

5 Transform and Quantization

Explanation of the triangle DCT/DST approch

The residual signal of the intra-inter predicted polygon is applied to the triangle transform DCT and DST. As shown in Figure 12, the texture polygon prediction residual is then transformed (DCT), quantized and bit compressed. The *i*-th polygon signal in *k*-th frame, $e^{(i)}(x,y,k)$, within the polygon shape is then shifted to the most left position and grouped to row vectors. Then the row vectors are transformed in horizontal-direction while using a 1-dimensional DCT of length N resulting in the corresponding vector with N transform coefficients each. Next the same procedure is repeated in vertical-direction. Those N_m elements of the row vectors which belong to the same column *m* are shifted to the most upper position and are formed into column vectors. Finally the column vectors are again transformed by a 1-dimensional DCT, but now in the vertical direction and with a transform length of N_m yielding the column vectors with the entire transform coefficients of the triangle DCT. The quantization is applied to the polygon unit transform coefficients. The same quantization method has been used to the polygon unit transform coefficients. (Further development shows the more effective quantization method could be proposed to improve the performance.) The zigzag scanning is proceeded to get the one-dimensional quantized transform coefficients.

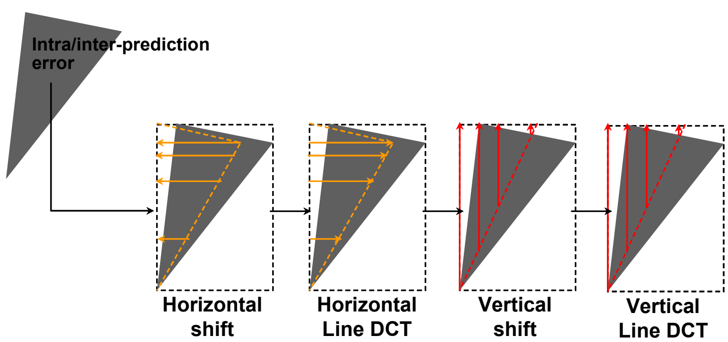


Figure 12. Polygon unit DCT / DST.

6 Entropy coding

Fig. 13 shows the final step of our proposed process. Coefficients are zigzag scanned into a one-dimensional array and CABAC entropy coding is applied the same way as in HEVC.

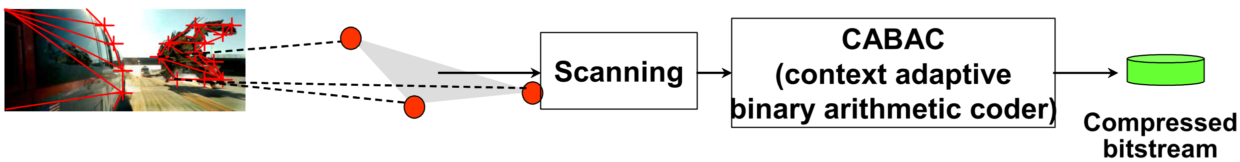


Figure 13. Scanning and CABAC encoding of the polygon-based texture unit approach.

7 Performance Evaluation

In order to discuss the compression performance of the proposed method compared to the existing H.264/AVC and HEVC codecs, we checked with BD-rate as the performance measure and the following tables shows that the proposed method achieves approximately 60% bit-saving on average compared to H.264/AVC and 20% bit-saving compared to HEVC. The parameters used in our implementation are listed below. Numbers of polygons per frame: 150~300, Motion vector search range for the affine transform: 4~8 pixel, Motion vector search precision: 1/4 pixel, Maximum polygon partitioning depth: 3, ㅎroup size of quantized DCT coefficients for CABAC: 16). The low delay main configuration option of HEVC was used.

BaketballPass_416x240_50.yuv

H.264	HEVC	Multiple-Texture
-	-32%	-58.9%

BlowingBubbles_416x240.yuv

H.264	HEVC	Multiple-Texture
-	-48%	-68.2%

BasketballDrill_832x480.yuv

H.264	HEVC	Multiple-Texture
-	-42%	-68.2%

8 Patent rights declaration(s)

Centri Alliance, Inc. may have current or pending patent rights 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).