ROTATE INTRA BLOCK COPY FOR STILL IMAGE CODING

Zhengdong Zhang, Vivienne Sze

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
{zhangzd, sze}@mit.edu

ABSTRACT

This paper proposes a method called *rotate intra block copy*, which extends the intra block copy technique by making the block matching process invariant to rotation. HEVC intra prediction plus rotate intra block copy gives an average of 20% reduction in residual energy (i.e. prediction error) compared to HEVC intra prediction plus intra block copy. As the motion vector correlation in rotate intra block copy is different from the intra block copy, a new method of motion vector coding is presented. The impact of angular resolution on residual energy reduction is also evaluated. In a full codec pipeline, this reduction in residual energy translates into a coding gain in BD-rate of 3.4% over HEVC intra prediction *plus* intra block copy for both screen content and camera-captured gray scale images.

Index Terms— Intra Prediction, Video Coding, Image Coding, High Efficiency Video Coding (HEVC), H.265, HEVC Screen Content Coding, HEVC Range Extensions

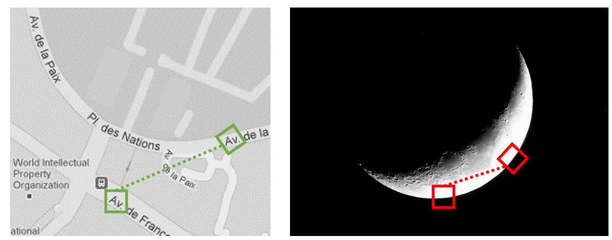
1. INTRODUCTION

In 2013, Facebook stored over 250 billion images, and over 350 million images were uploaded daily [1]. As the number of images stored and transmitted continues to grow, there is a strong need for better intra coding tools.

High Efficiency Video Coding (HEVC) [2, 3] is the latest video coding standard that delivers significant coding gains for both video [4] and still images [5]. HEVC's *Still Image Profile*, which only uses intra coding tools [6], gives 43.6% coding gains over JPEG.

There are several new intra coding tools that are under consideration for the Screen Content Coding extension of HEVC [7]. Screen content, such as maps, text and graphics, have different characteristics from camera-captured images with patterns that tend to be clean and repetitive in texture. A technique called intra block copy [8, 9, 10] exploits this property for coding gain, by finding a matching block that can be used as a predictor.

However, images can also contain many blocks that can share common patterns but with different orientations. For



(a) map (b) moon

Fig. 1: Rotated patch matching. Left: matched texts on screen content with different orientation. Right: matched blocks along the curved boundary of an object.

example, in the screen content such as maps, as shown in Fig. 1(a), there can be rotated repeated text. Blocks on the same object boundary tend to also contain rotated repeated patterns, as shown in Fig. 1(b). Unfortunately, due to this orientation difference, such blocks cannot be used as predictors by intra block copy for coding gains.

Accordingly, this paper proposes extending the intra block copy so that the block matching process is invariant to rotation using a technique called *rotate intra block copy*. As a result, the residual energy (i.e. prediction error) is substantially reduced, which can result in coding gains.

This paper presents the following contributions for rotate intra block copy:

- a method to rotate the predictor to reduce the residual energy
- a method of coding the rotated motion vector
- an analysis of the impact of angular resolution on residual energy
- an evaluation of the coding gains relative to HEVC intra prediction plus intra block copy in a full codec

2. INTRA PREDICTION WITH ROTATE INTRA BLOCK COPY

The encoder processes blocks of pixels of an image in a raster scan order. To encode a target block $P(x_0) \in \mathbb{R}^{n\times n}$ with top left corner located at x_0, the encoder can use any previously coded pixels before $P(x_0)$, denoted by $\Omega(x_0)$, within the same frame as a predictor as shown in Fig. 2. To predict

This project is supported by the SAMSUNG Global Research Outreach (GRO) Program.

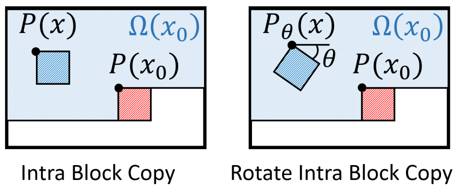


Fig. 2: Intra block copy vs. rotate intra block copy.

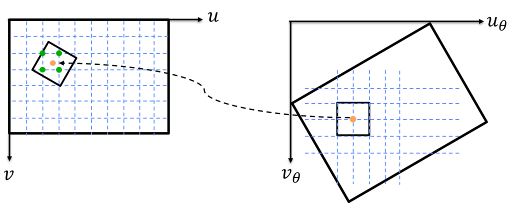


Fig. 3: Rotating an image with bilinear interpolation.

$P(x_0)$, intra block copy method seeks a block anchored at x within $\Omega(x_0)$ that minimizes the sum of squared differences (SSD)

$$\min_{x\in\Omega(x)} \|P(x) - P(x_0)\|_F^2 \tag{1}$$

Here $\|X\|_F = \sqrt{\sum_{i,j} X_{i,j}^2}$ denotes the Frobenius norm of a matrix. Rotate intra block copy adds additional flexibility on top of Eq. (1) by allowing the block to rotate by an arbitrary angle θ. Let $P_\theta(x)$ represent the block rotated by θ which starts from x. Then the search task can be formulated as

$$\min_{x\in\Omega(x),\theta} \|P_\theta(x) - P(x_0)\|_F^2 \tag{2}$$

Note that $P_\theta(x)$ is always represented by an axis aligned matrix despite its rotation. To determine the intensity of each pixel in $P_\theta(x)$, we apply the standard image transformation technique illustrated in Fig. 3. Specifically, it first establishes the rotating mapping between the coordinate system of the rotated patch $P_\theta(x)$ and the coordinate system of the source image. This mapping projects each pixel in $P_\theta(x)$ back to the source image. Since the mapped points may be fractional, bilinear interpolation is performed to compute their intensity values.

The optimization in Eq. (2) is continuous and highly non-convex because of θ; to make it tractable, the search rotation angle θ is restricted to a discrete list of candidates, 0:(π/N):π, leaving N as a parameter to tune. Sec. 4 describes how to select N.

The optimal solution is found through an exhaustive search by enumerating each candidate θ, finding x given the fixed θ, repeating this for all candidates θ, and selecting the best (x,θ) pair overall. Given the candidate θ, the search for x that minimizes the objective in Eq. (2) can be done in two ways (Fig. 4):

1. rotate the target block by $-\theta$ and search for a rotated block on the original axis-aligned image

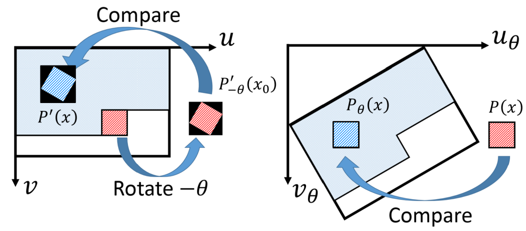


Fig. 4: **Left:** Rotate the target block and search on the original image. Note that the rotated patch is padded to a larger rectangular patch and the padded area should be excluded for comparison during the search. **Right:** Rotate the image and search for for an axis aligned block.

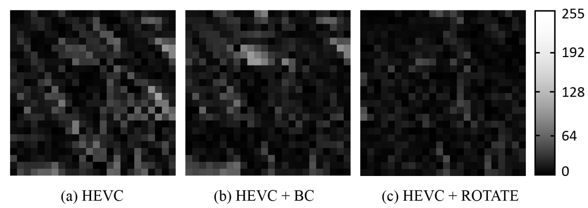


Fig. 5: Absolute residual from different intra prediction methods for PartyScene (cropped).

2. rotate the original image by θ and search for an axis-aligned block

The first approach is trying to minimize the SSD between $P'(x)$ and $P'_{-\theta}(x_0)$, which is not equivalent to the actual objective function in Eq. (2) due to the interpolation introduced by rotation. This mismatch can affect the rate-distortion optimization in a codec. In practice, we observe that the SSD between the target patch and the optimal patch found increases by 1% if we adopt the first scheme. Therefore, the second method is used to avoid this unnecessary loss of accuracy.

Fig. 5 visualizes the residual generated by the HEVC intra prediction, HEVC plus intra block copy (HEVC+BC) and HEVC plus rotate intra block copy (HEVC+ROTATE) on PartyScene from the JCT-VC common test sequences. The residual of the rotate intra block copy is visually sparser compared with the other two baseline methods. This represents a 40% and 27% reduction in residual energy (i.e. SSD) as compared with the original HEVC intra prediction and HEVC plus intra block copy, respectively.

3. MOTION VECTOR CODING

Suppose x^* is the location of the optimal patch we find to predict the target patch located at x_0. To signal x^* to the decoder, we compute and signal the motion vector similar to the intra block copy. Since the search for x^* is performed on the grid of the rotated coordinate system (u_θ, v_θ), x^* is an integer in (u_θ, v_θ) but fractional in the source coordinate system (u, v). To avoid encoding fractional numbers, we choose to encode x^* in (u_θ, v_θ) instead of (u, v). Note that x_0 is in the original unrotated coordinate system, for it to help the prediction of

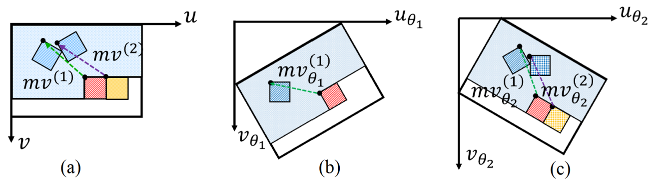


Fig. 6: Rotate motion vector prediction

x^* we need to rotate it to (u_θ, v_θ) by multiplying it with the rotation matrix R_θ with rotate angle θ defined as

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3}$$

Since $R_\theta x_0$ may be fractional, we need to perform rounding on it. Let us denote the output by $\text{round}\,(R_\theta x_0)$. Then we define the motion vector in the rotated coordinate system (u_θ, v_θ) as $mv = x^* - \text{round}\,(R_\theta x_0)$. The encoder computes mv and signals it together with θ. At the decoder side, mv, x_0 and θ are all known so the decoder can retrieve $x^* = mv + \text{round}\,(R_\theta x_0)$.

For intra block copy, the motion vectors of neighboring blocks are correlated, and this correlation is exploited to reduce signaling cost. However, with rotate intra block copy, the rotation angle of the neighboring blocks can be different, which causes the motion vectors to no longer be correlated as shown in Fig. 6 and explained later. This can be addressed by transforming the motion vectors of neighboring block to the same coordinate system to increase correlation.

To be specific, suppose there are two adjacent target blocks with rotation angle θ_1, θ_2 and motion vector $mv^{(1)}$ and $mv^{(2)}$, as shown in Fig. 6(a). The motion vector for the first block $mv^{(1)}$ is encoded as $mv^{(1)}_{\theta_1}$ in the rotated coordinate system $(u_{\theta_1}, v_{\theta_1})$ (Fig. 6(b)). Similarly $mv^{(2)}$ is encoded as $mv^{(2)}_{\theta_2}$ in $(u_{\theta_2}, v_{\theta_2})$. Although $mv^{(1)}$ is close to $mv^{(2)}$, there is a big gap between $mv^{(1)}_{\theta_1}$ and $mv^{(2)}_{\theta_2}$ due to different rotation angle θ_1 and θ_2. To compensate, we have to rotate $mv^{(1)}_{\theta_1}$ from $(u_{\theta_1}, v_{\theta_1})$ to $(u_{\theta_2}, v_{\theta_2})$(Fig. 6(c)). This is achieved by multiplying $mv^{(1)}_{\theta_1}$ with a rotation matrix $R_{\theta_2 - \theta_1}$.

The motion vector predictor is generated based on the rotated motion vector of the neighboring block.[1] If the previously encoded block is not rotate intra block copy, the predictor is set to zero. Motion vector prediction can be adaptively enabled using a flag. If enabled the motion vector difference is encoded, while if disabled the motion vector is explicitly encoded. Both the motion vector difference and the motion vector are encoded with a third order Exp-Golomb codeword.

We collect the statistics of the motion vector codes when encoding class C sequences of JCTVC test sequences. The results suggest that motion vector prediction is enabled on

[1]Top neighbor if first block in row, otherwise left neighbor.

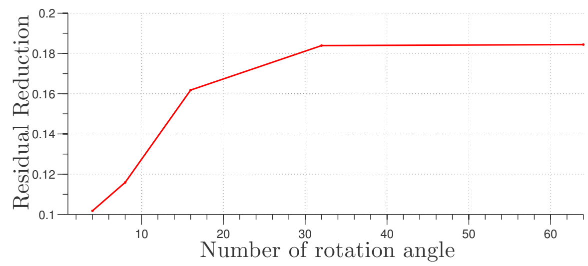


Fig. 7: Tradeoff between angular resolution and residual energy

24.5% of the blocks, validating the need for a flag to dynamically enable motion vector prediction for each coding unit (CU). In addition, for the blocks where motion vector prediction are enabled, the rotating motion vector compensation technique reduces the average bit rates for the motion vector difference by 25%, which proves its effectiveness.

4. ANGULAR RESOLUTION OF ROTATION

The angular resolution for the search has a significant impact on both the speed and coding performance of the encoder. Higher angular resolution increases the chance of locating a more accurate match for the target block; however, it also slows down the searching process and requires more bits to represent the rotation angle.

The tradeoff between these factor was evaluated with several different angular resolutions across several test images[2] as shown in Fig. 7. Step size of $\pi/32$ is selected for favorable balance between coding gain and encoding speed.

Rather than sending the actual rotation angle, a list is constructed containing all supported rotation angles, and the index of the rotation angle is transmitted. For a step size of $\pi/32$, the rotation index is coded with a 6-bit[3] fixed length codeword. No correlation was observed between rotation angles of neighboring blocks.

5. CODING GAIN EVALUATION

5.1. Setup of test codec

A complete codec was implemented in MATLAB to evaluate how the reduction in residual energy impacts coding efficiency. The pipeline of the codec consists of intra prediction, transform-quantization, and entropy coding.

The intra prediction module consists of HEVC intra prediction (with DC, planar and 33 angular prediction directions, but without intra smoothing), intra block copy and rotate intra block copy. The HEVC discrete cosine transform (DCT) and quantization is applied to the prediction error to compute the residual[11]. The residual is then encoded using a zigzag scan and run-length-encoding (rle). The run and length values are mapped to codewords using an optimal Huffman table based

[2]First frame of BasketballPass, BQSquare, BasketballDrillText

[3]$0{:}(\pi/32){:}\pi$ contains 33 values, requiring 6 bits to encode instead of 5

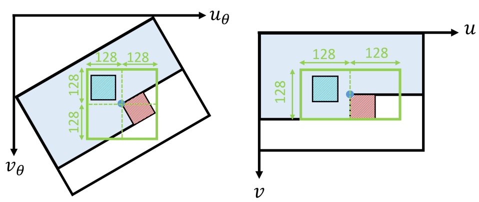


Fig. 8: Restriction on the search range

on rle sequences of all blocks. The codec also supports an HEVC style quad-tree block structure with coding units (CU) from 16×16 down to 4×4 block sizes. Each prediction unit (PU) can either be HEVC intra prediction, intra block copy or rotate intra block copy (if enabled).

To signal the encoded image requires the following bits for various sets of information including

- *Picture*: height, width, QP, Coding Tree Unit (CTU) size, max CU depth, encoding mode (HEVC only, HEVC+BC, HEVC+ROTATE), rotation range and resolution
- *Block Structure and Prediction*: split flag, enable intra block copy or rotate intra block copy, HEVC intra prediction mode, motion vector info (motion vector predictor enable flag, motion vector/motion vector difference), rotation angle index
- *Residual*: run, length

The rate distortion optimization is performed using the standard objective function $D + \lambda \cdot R$ where D is the quantization error of a block, and R is the estimated bits of the block including residual bits, motion vector info bits, and intra prediction mode/rotation angle index bits. The residual bits are estimated by multiplying the number of run-length values with the average codeword length in the rle sequence, which is learnt from the test results.

The same exhaustive intra motion vector search strategy is applied for both intra block copy and rotate intra block copy with the same motion vector coding to make the comparison fair (for intra block copy, no neighbor rotation is needed). For rotate intra block copy, the search for the motion vector is limited to the area within ± 128 pixels around the target patch along both axes (Fig. 8(a)). A similar restriction is applied on intra block copy, but the vertical motion vector component can only be negative (Fig. 8(b)).

5.2. Test Results

Table 1 shows the coding efficiency of HEVC intra prediction plus intra block copy (HEVC+BC) and HEVC intra prediction plus rotate intra block copy (HEVC+ROTATE) evaluated using the first frame of the test sequences in the JCT-VC common test conditions[12][4]. The BD-rate[13] is measured for QP={22, 27, 32, 37} and only the luma component is considered. For HEVC+ROTATE, the encoder performs both HEVC and rotate intra block copy predictions and picks the best from the rate-distortion optimization; the same goes for HEVC+BC. It can be seen that HEVC+ROTATE achieves on average residual energy reduction of 20% reduction compared with HEVC+BC. With the proposed codec, this reduction translates into a 3.4% coding gain in BD-rate averaged over all the test sequences. Note that the bits for block structure and prediction actually increase by 26% to signal the rotate angle, but those bits only account for 27% of the total bits and their increase is surpassed by the reduction of the residual bits and as a whole the proposed technique achieves the aforementioned coding again.

	Sequence	Residual reduction	BD-rate
Class C	RaceHorse	23.66%	-4.54
	PartyScene	27.64%	-4.45
	BQMall	17.92%	-2.63
	BasketballDrill	22.12%	-3.40
Class D	BQSquare	30.82%	-4.99
	BasketballPass	15.44%	-1.84
	BlowingBubbles	7.59%	-2.81
	RaceHorse	28.97%	-4.42
Class E	FourPeople	18.09%	-2.54
	Johnny	12.79%	-2.35
	KristenAndSara	15.67%	-2.43
Class F	BasketballDrillText	21.15%	-3.64
screen	SlideShow	29.01%	-7.43
content	SlideEditing	19.12%	-0.74
Class C Average		22.83%	-3.76
Class D Average		20.70%	-3.52
Class E Average		15.52%	-2.44
Class F Average		23.09%	-3.94
Overall Average		20.71%	-3.44

Table 1: Residual energy reduction and BD-rate of HEVC+ROTATE compared with HEVC+BC.

6. CONCLUSION

In this paper, we proposed a novel intra prediction method that delivers significant residual energy reduction compared with existing intra coding tools such as HEVC intra prediction and the intra block copy technique. We also presented an efficient way to encode the rotated motion vector and determined an angular search resolution that gives a favorable search vs. coding efficiency tradeoff. We conducted experiments on common test sequences to measure how the reduction in residual energy can be translated into coding gains.

[4]The Huffman dictionary is not included in the bits for HEVC+BC or HEVC+ROTATE. This is because on average the bits needed to encode the Huffman dictionary of HEVC + ROTATE (2153 bits) is smaller than the other (2310 bits). We exclude this artificial gain in BD-rate because the actual HM pipeline is not using Huffman table for entropy coding.

7. REFERENCES

[1] "A Focus on Efficiency: A whitepaper from Facebook, Ericsson and Qualcomm," Tech. Rep., internet.org.

[2] *High efficiency video coding*, ITU-T Recommendation H.265 and ISO/IEC 230082, April 2013.

[3] V. Sze, M. Budagavi, and G. J. Sullivan, Eds., *High Efficiency Video Coding (HEVC): Algorithms and Architectures*, Integrated Circuits and Systems. Springer, 2014.

[4] J. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding StandardsIncluding High Efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 22, no. 12, pp. 1669–1684, Dec 2012.

[5] T. Nguyen and D. Marpe, "Performance analysis of HEVC-based intra coding for still image compression," in *Picture Coding Symposium (PCS)*, 2012, pp. 233–236.

[6] J. Lainema and W.-J. Han, "Intra-Picture Prediction in HEVC," in *High Efficiency Video Coding (HEVC): Algorithms and Architectures*, V. Sze, M. Budagavi, and G. J. Sullivan, Eds. Springer, 2014.

[7] G.J. Sullivan, J.M. Boyce, Ying Chen, J.-R. Ohm, C.A. Segall, and A. Vetro, "Standardized Extensions of High Efficiency Video Coding (HEVC)," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1001–1016, Dec 2013.

[8] M. Budagavi and D.-K. Kwon, "JCTVC-M0350: AHG8: Video coding using Intra motion compensation," Joint Collaborative Team on Video Coding (JCT-VC), April 2013.

[9] S. L. Yu and C. Chrysafis, "JVT-C151r1: New intra prediction using intra-macroblock motion compensation," Joint Video Team (JVT), May 2002.

[10] C. Pang, J. Sole, L. Guo, M. Karczewicz, and R. Joshi, "JCTVC-N0256: Non-RCE3:Intra motion compensation with 2-D MVs," Joint Collaborative Team on Video Coding (JCT-VC), July 2013.

[11] M. Budagavi, A. Fuldseth, and G. Bjntegaard, "HEVC Transform and Quantization," in *High Efficiency Video Coding (HEVC): Algorithms and Architectures*, V. Sze, M. Budagavi, and G. J. Sullivan, Eds. Springer, 2014.

[12] F. Bossen, "JCTVC-L1100: Common HM test conditions and software reference configurations," Joint Collaborative Team on Video Coding (JCT-VC), Jan 2013.

[13] G. Bjøntegaard, "VCEG-M33: Calculation of Average PSNR Differences between RD curves," Video Coding Experts Group (VCEG), April 2001.