Discrete Cosine Basis Oriented Motion Modeling for Fisheye and 360 Degree Video Coding

Ashek Ahmed & Manoranjan Paul
School of Computing and Mathematics, Charles Sturt University, Australia.
{aahmmed, mpaul}@csu.edu.au

Abstract

Motion modeling plays a central role in video compression. Discrete cosine basis has the ability to efficiently model complex motion fields. In this work, we investigate the motion modeling behaviour of the discrete cosine basis equipped with higher frequency cosine vectors for fisheye and 360-degree video frames. In particular, the developed discrete cosine basis is used as a single high-order model to describe a frame's motion; we employ this motion to produce an extra prediction reference, which is added to the HEVC list of references. Experimental results show a reduction in bit rate of up to 7.91% for fisheye video sequences and up to 2.05% for 360-degree video sequences, when compared to standalone HEVC.

The Discrete Cosine Basis for Motion

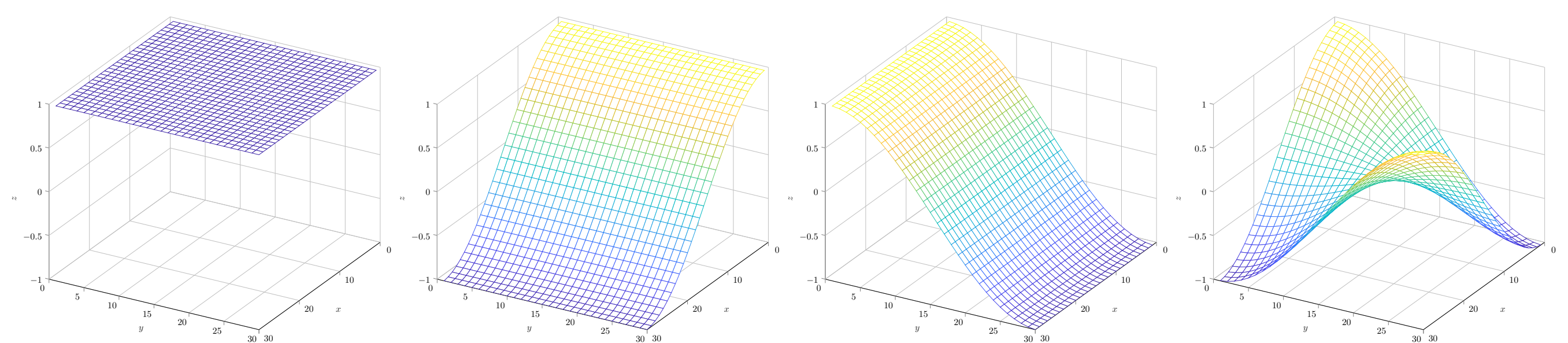

Figure 1: The two-dimensional cosine vectors used in [1] to represent motion; from left to right, the plots are for $\mathbf{u} = (0,0)$, $(1,0)$, $(0,1)$, and $(1,1)$.

A two-dimensional vector $\phi_{\mathbf{u}}$ in the 2D discrete separable cosine basis can be characterized by $\mathbf{u} = (u_1, u_2)$, where $u_1 \in \{0, 1, \dots\}$ and $u_2 \in \{0, 1, \dots\}$ represent, respectively, the horizontal and vertical frequencies of this vector. This vector is evaluated, at location $\mathbf{x} = (x_1, x_2)$ of the frame under consideration, using

$$\phi_{\mathbf{u}}(\mathbf{x}) = \cos\left(\frac{(2x_1+1)\pi u_1}{2W}\right) \cdot \cos\left(\frac{(2x_2+1)\pi u_2}{2H}\right) \tag{1}$$

where W and H are the width and height of the frame, respectively. Then, the motion vector $\mathbf{v} = (v_1, v_2)$ at location $\mathbf{x}$ is obtained from

$$v_1(\mathbf{x}) = \sum_{\mathbf{u} \in \mathbf{U}} m_{1,k} \phi_{\mathbf{u}}(\mathbf{x}) \tag{2}$$

$$v_2(\mathbf{x}) = \sum_{\mathbf{u} \in \mathbf{U}} m_{2,k} \phi_{\mathbf{u}}(\mathbf{x}) \tag{3}$$

where $\{m_{1,k}, m_{2,k}\}_k$ are the parameters of the model. Fig. 1 shows the corresponding cosine vectors used in the work [1].

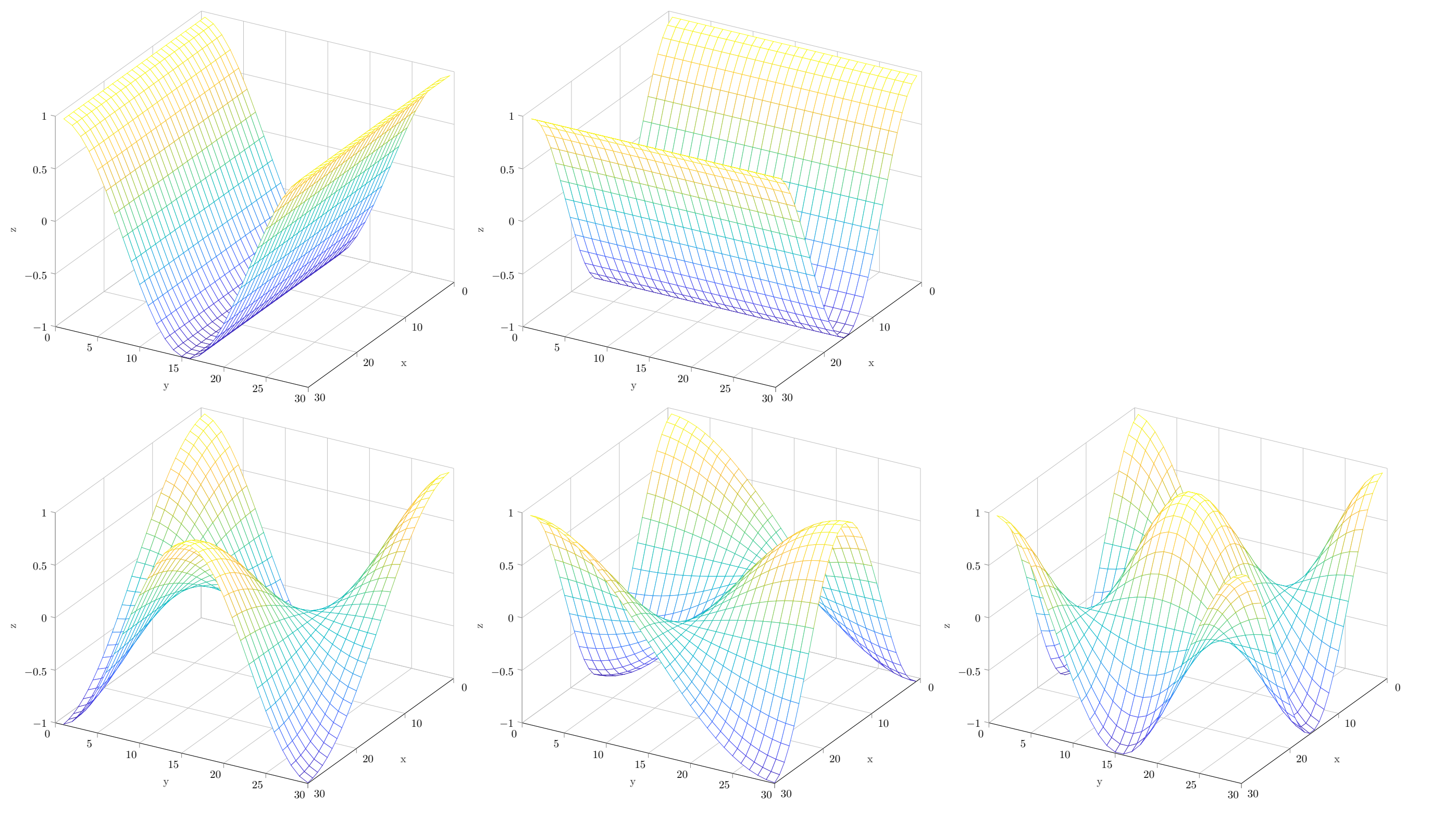

Figure 2: Top: the two-dimensional cosine vectors used in this work to represent motion; from left to right, the plots are for $\mathbf{u} = (0,2)$, and $(2,0)$. Bottom: from left to right, the plots are for $\mathbf{u} = (1,2)$, $(2,1)$, and $(2,2)$.

Letting the horizontal and vertical frequencies of considered cosine vectors to be $u_1 \in \{0, 1, 2\}$ and $u_2 \in \{0, 1, 2\}$, would incorporate additional cosine vectors in the discrete cosine basis. Fig. 2 shows these new vectors which are used in this work along with the vectors of Fig. 1.

Prediction using the Proposed Motion Model

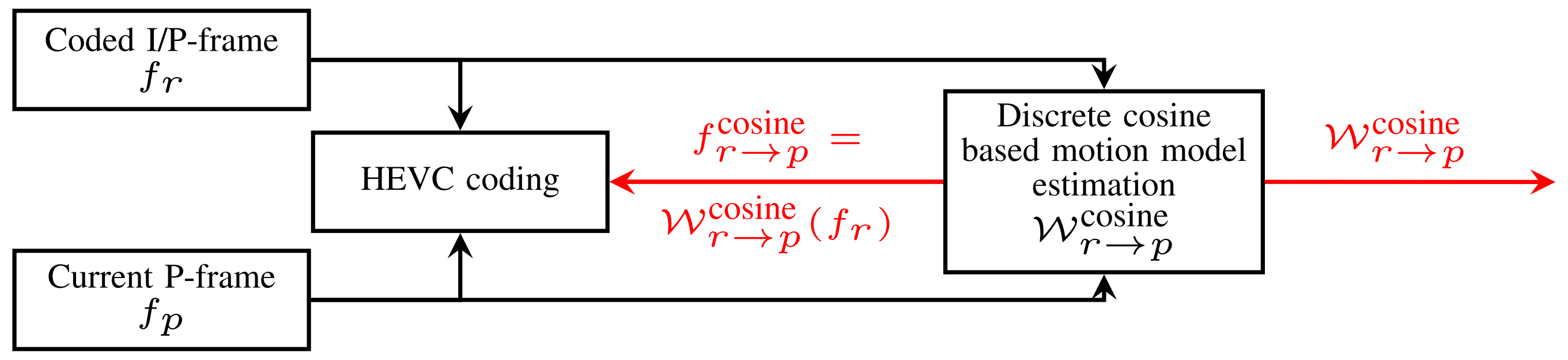

Figure 3: Block diagram showing the discrete cosine-based prediction generation process at the encoder.

The parameters of the proposed motion model are estimated per frame basis. After estimation, these parameters are employed by the encoder and decoder to generate an additional reference frame. We write $\mathcal{W}^{\text{cosine}}_{r \to p}$ for the motion compensation operator that associate locations in the frame being predicted f_p with locations in its reference frames f_r, obtained using these motion parameters. This way, the additional reference frame $f^{\text{cosine}}_{r \to p}$ is obtained using

$$f^{\text{cosine}}_{r \to p} = \mathcal{W}^{\text{cosine}}_{r \to p}(f_r) \tag{4}$$

Fig. 3 shows a simplified block diagram of the proposed encoding architecture. In this work, every P-frame f_p, has an additional reference frame, obtained using the proposed motion model $\mathcal{W}^{\text{cosine}}_{r \to p}$.

Experimental Analysis

Figure 4: Frames from the fisheye video sequences [4] used in this work. The sequences left-to-right are: LectureB, DriveE, DriveB.

The rate-distortion (RD) performance of the employed coder is investigated, at first on 3 different fisheye video sequences which are publicly available and part of the data set in [4]; frames from these sequences are shown in Fig. 4. Frames from each 1080×1080 fisheye sequence are coded by the HM 16.10 reference software for HEVC. The HM encoder is configured using the low delay P- GOP structure i.e. IPPP. . . P as per the common test conditions [3]. Four different quantization parameter values (QP = 22, 27, 32, 37) are used. For each current P-frame, the already coded I- or P-frame, i.e. the typical reference in HEVC, is used to estimate the motion parameters $\mathcal{W}^{\text{cosine}}_{r \to p}$. These parameters are then employed to generate the prediction $f^{\text{cosine}}_{r \to p}$ which is inserted into LIST0 to be used as an additional reference frame. Table 1 tabulates the Bjøntegaard Deltas [2] from this modified HEVC codec over the considered test sequences, when compared to standalone HEVC.

Sequence	Delta rate [1]	Delta rate
LectureB	-6.51%	-7.91%
DriveE	-4.40%	-5.21%
DriveB	-3.27%	-3.81%

Table 1: The Bjøntegaard delta gains obtained for the fisheye test sequences over standalone HEVC when the discrete cosine-based reference is employed.

Figure 5: Frames from the 360-degree video sequences used in this work. The sequences left-to-right are: Lion, Shark-encounter.

Due to the similarities between fisheye and 360 degree video frames, next we propose to predict the motion in 360 degree video sequences, shown in Fig. 5, employing the modified discrete cosines based motion model $\mathcal{W}^{\text{cosine}}_{r \to p}$. Table 2 tabulates the Bjøntegaard Deltas [2] for these two test sequences and Fig. 6 shows a comparative analysis between typical HEVC encoder and the proposed modified HEVC encoder on prediction unit (PU) structure level.

Sequence	Delta rate	Delta PSNR
Lion	-1.29%	0.07 dB
Shark-encounter	-2.05%	0.09 dB

Table 2: The Bjøntegaard delta gains obtained for the 360-degree test sequences over standalone HEVC when the discrete cosine-based reference is employed.

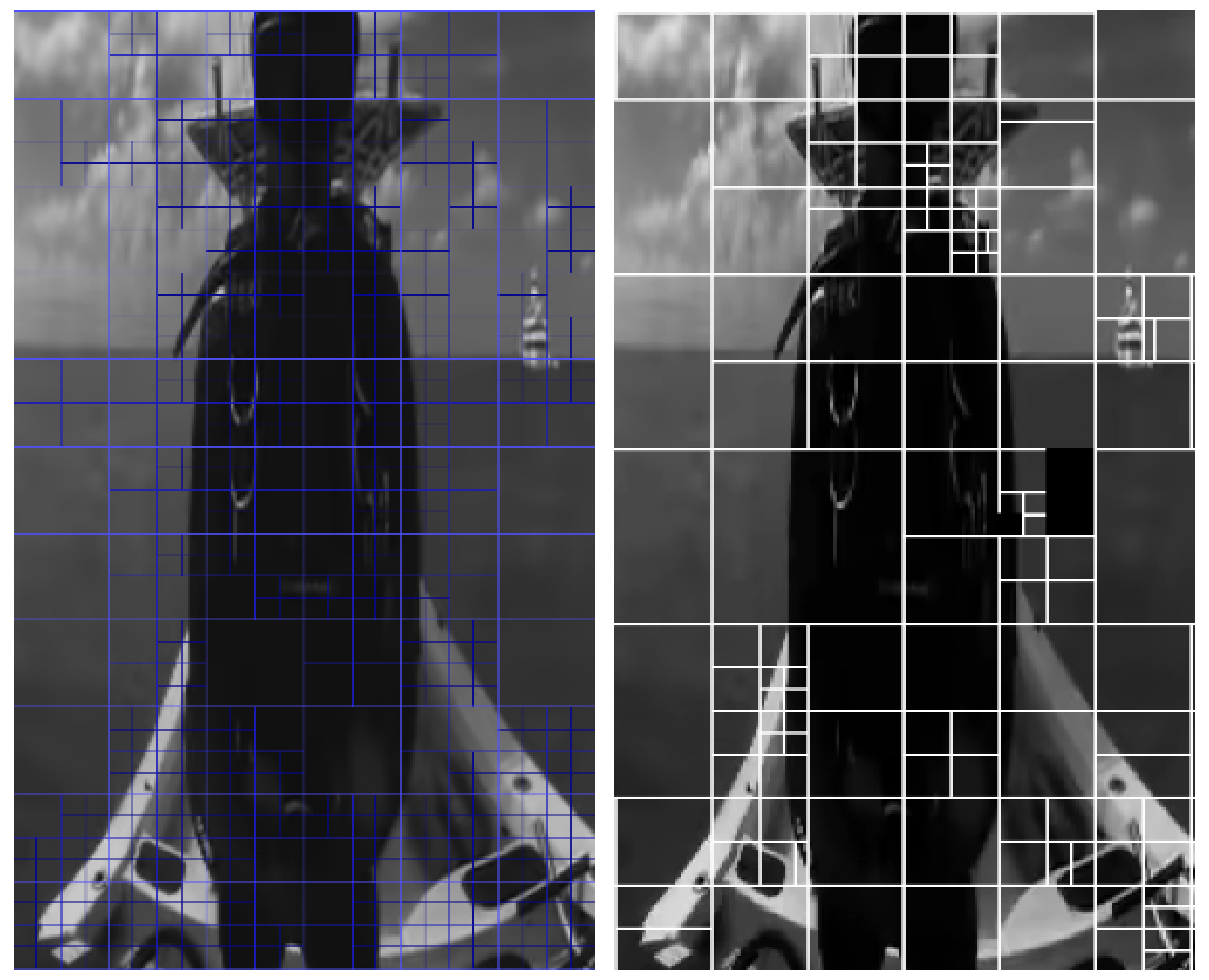

Figure 6: Prediction unit (PU) structure for frame 5 of the *Shark-Encounter* 360 degree video sequence produced by: (from left-to-right) (a) the standard HM encoder and (b) by the proposed modification to the HM.

References

[1] A. Ahmed, M. M. Hannuksela, and M. Gabbouj. Fisheye video coding using elastic motion compensated reference frames. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2027–2031, Sep. 2016.

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

[3] F. Bossen. Common test conditions and software reference configurations. In *document JCTVC-H1100, JCT-VC*, San Jose, CA, Feb 2012.

[4] A. Eichenseer and A. Kaup. A data set providing synthetic and real-world fisheye video sequences. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, March 2016.