Leveraging the Discrete Cosine Basis for Better Motion Modelling in Highly Textured Video Sequences

Ashek Ahmed, Aous Naman, and Mark Pickering

The University of New South Wales, Australia.

Abstract

Motion modelling plays a central role in video compression. This role is even more critical in highly textured video sequences, whereby a small error can produce large residuals that are costly to compress. In this work, we explore the use of the discrete cosine basis for motion modelling in highly textured video sequences, and show that this is beneficial. In particular, we use 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 that a median delta bit rate of 4.44% is achievable over conventional HEVC if this extra reference frame is used in addition to the temporal references offered by HEVC.

The Discrete Cosine Basis for Motion

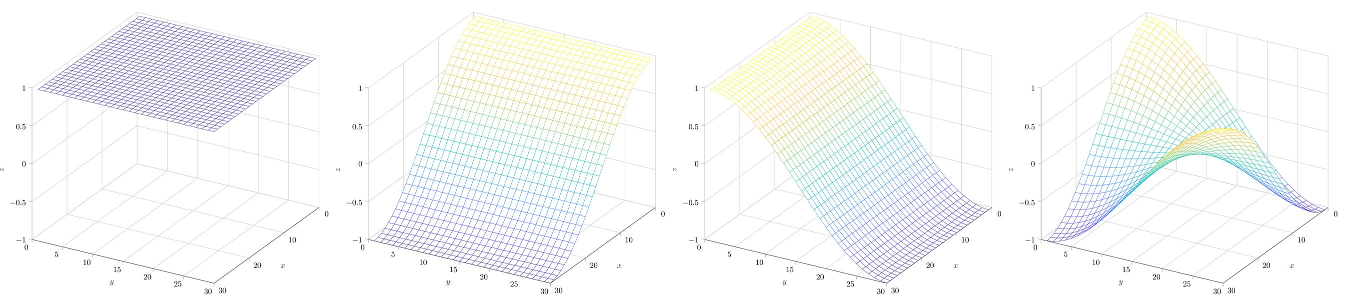

Figure 1: The two-dimensional cosine vectors used in this work to represent motion; from left to right, the plots are for $\mathbf{u} = (0,0)$, $(1,0)$, $(0,1)$, and $(1,1)$.

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 cosine vectors used in this work. While the vectors associated with $\mathbf{u} = (0,0)$, $(1,0)$, and $(0,1)$ are very similar to the affine vectors, the cosine vector associated with $\mathbf{u} = (1,1)$ has a higher order. The parameters $\{m_{1,k}, m_{2,k}\}_{k=0}^{3}$ are estimated using gradient-based image registration techniques.

Prediction using the Proposed Motion Model

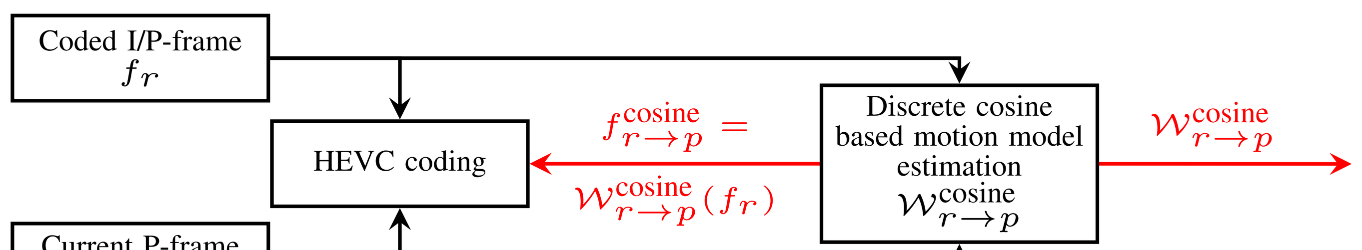

Figure 2: Block diagram showing the discrete cosine-based prediction generation process at the encoder.

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. 2 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

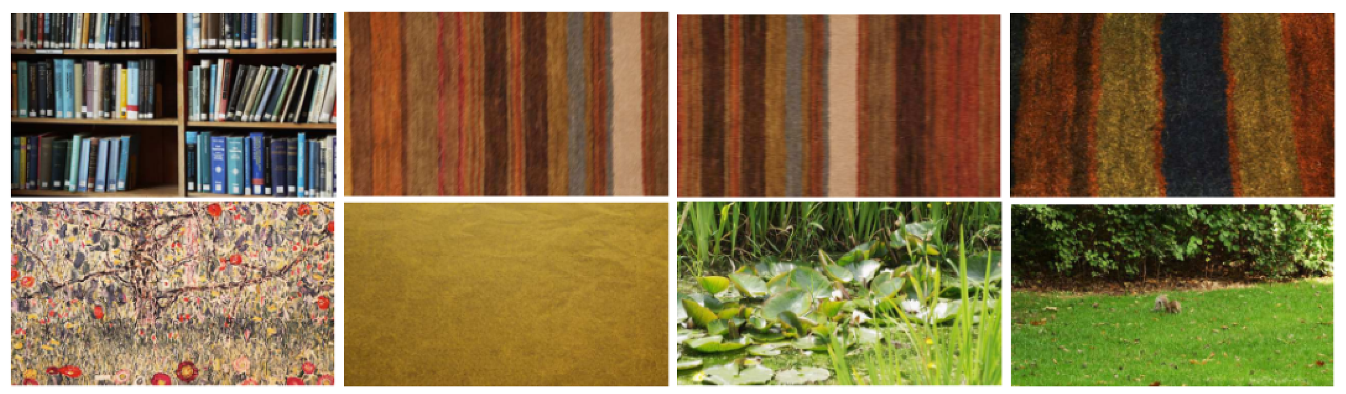

Figure 3: Frames from the highly textured video sequences used in this work. The sequences left-to-right, top-to-bottom, are: Bookcase, CarpetCircleFast, CarpetPanAverage, CarpetSlowTrans, PaintingTilting, PaperStatic, PondDragonFlies, Squirrel.

The rate-distortion (RD) performance of the employed coder is investigated, on 8 different texture video sequences which are publicly available and part of the data set *BVI Texture*; frames from these sequences are shown in Fig. 3. The first 300 frames of each 1080p 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. Four different quantization parameter values (QP = 22, 27, 32, 37) are used. For each P-frame, the available I- or P-frame is used to estimate the 8 parameters of the discrete cosine-based motion model $\mathcal{W}^{\text{cosine}}_{r\to p}$. The fractional part of these parameters is limited to $1/64$, and they are coded using the Exponential Golomb coding technique. The frame, $f^{\text{cosine}}_{r\to p}$, is inserted as a reference frame into LIST0, which is tweaked such that $f^{\text{cosine}}_{r\to p}$ becomes the first reference frame, ahead of the usual reference f_r.

Sequence	Delta rate	Delta PSNR
Bookcase	-4.12%	$+0.17$ dB
CarpetCircleFast	0.51%	-0.01 dB
CarpetPanAverage	-2.24%	$+0.06$ dB
CarpetSlowTrans	-7.23%	$+0.17$ dB
PaintingTilting	**-43.65%**	**$+1.84$** dB
PaperStatic	-3.82%	$+0.10$ dB
PondDragonflies	-4.91%	$+0.19$ dB
Squirrel	-4.76%	$+0.10$ dB

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

Fig. 4 shows the rate distortion curve for the *PaintingTilting* sequence while Table 1 tabulates the Bjøntegaard Deltas for all test sequences under consideration. The employed hybrid prediction paradigm generates a bit rate saving for all test sequences other than the *CarpetCircleFast* sequence; this sequence contains fast motion which is difficult to model. When the same texture scene undergoes different motions, namely in the sequences *CarpetPanAverage* and *CarpetSlowTrans*, delta bit rates of 2.24% and 7.23% can be achieved, respectively. The maximum gain is obtained in the *PaintingTilting* sequence; this sequence contains complex texture whose underlying motion is very difficult to model using the translational motion model of HEVC. The discrete cosine-based motion model manages to capture this complex motion to a significant extent. Fig. 5 and 6 present a comparitive analysis in terms of the prediction unit structure employed by the standalone and modified HEVC encoders respectively.

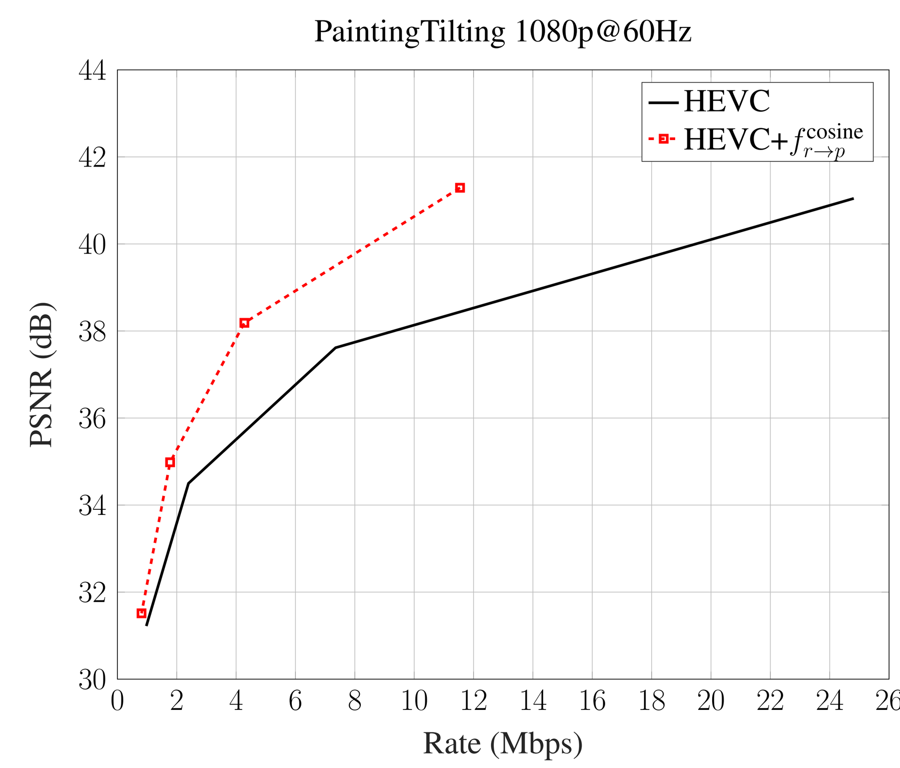

Figure 4: Rate distortion performance of two different coding strategies for the *PaintingTilting* (1920×1080) texture video sequence. A bit saving of 43.65% is achieved by using the discrete cosine-based reference $f^{\text{cosine}}_{r\to p}$ in addition to the usual reference f_r.

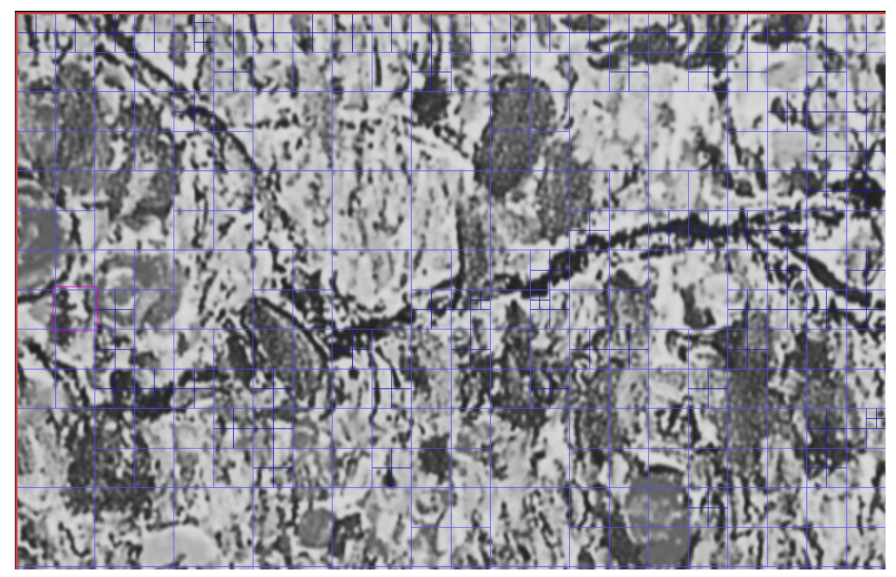

Figure 5: Prediction unit (PU) structure for frame 5 of the 1080p *PaintingTilting* textured video sequence [?] produced by the HM encoder.

Figure 6: Prediction unit (PU) structure for frame 5 of the 1080p *PaintingTilting* textured video sequence [?] produced by the proposed modification to the HM encoder.