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 modest delta bit rate of 4.4% 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
A two-dimensional vector $\mathbf{u}$ in the 2D discrete separable cosine basis can be characterized by $u = (u_0, u_1)$, where $u_0 \in \{0, 1, 2, \ldots \}$ and $u_1 \in \{0, 1, 2, \ldots \}$ represent, respectively, the horizontal and vertical frequencies of this vector. This vector is evaluated, at location $x$ = $(x_0, x_1)$ of the frame under consideration, using

$$u_0(x) = \cos \left( \frac{2\pi x_0 + 1}{2H} u_0 \right) \cos \left( \frac{2\pi x_1 + 1}{2L} u_1 \right)$$

where $H$ and $L$ are the width and height of the frame, respectively. Then, the motion vector $v(x, y)$ at location $x$ is obtained from

$$v(x, y) = \sum_{u_0, u_1} u_0(x) u_1(y)$$

where $(u_0, u_1)$ are the parameters of the model.

Prediction using the Proposed Motion Model
![Diagram](Image)

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 $W_{\text{cosp}}$ for the motion compensation operator that associate locations in the frame being predicted $f_p$ with locations in its reference frames $f_x$, obtained using these motion parameters. This way, the additional reference frame $f_{\text{cosp}} = W_{\text{cosp}} f_p$ is obtained using

$$f_{\text{cosp}}(x) = W_{\text{cosp}} f_p(x).$$

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 $W_{\text{cosp}} f_p$.

Experimental Analysis
![Diagram](Image)

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

The rate-distortion (RD) performance of the employed coder is investigated, on 5 different texture video sequences which are publicly available and part of the data set BT1 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. IPPPP・・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 1- or P-frame is used to estimate the 5 parameters of the discrete cosine-based motion model $W_{\text{cosp}}$. The fractional part of these parameters is limited to 1/64, and they are coded using the Exponential Golomb coding technique. The frame, $f_{\text{cosp}}$, is inserted as a reference frame into List0, which is tweaked such that $f_{\text{cosp}}$ becomes the first reference frame, ahead of the usual reference $f_x$. The Bjøntegaard delta gains obtained for the texture video test sequences over standalone HEVC when the discrete cosine-based reference is employed.

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

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Delta rate</th>
<th>Delta PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookcase</td>
<td>-4.12%</td>
<td>+0.17 dB</td>
</tr>
<tr>
<td>CarpetCircleFast</td>
<td>0.51%</td>
<td>-0.61 dB</td>
</tr>
<tr>
<td>CarpetPavAverage</td>
<td>-2.21%</td>
<td>+0.06 dB</td>
</tr>
<tr>
<td>CarpetSlowTrans</td>
<td>-7.23%</td>
<td>+0.17 dB</td>
</tr>
<tr>
<td>PaintingTilting</td>
<td>-43.65%</td>
<td>+1.84 dB</td>
</tr>
<tr>
<td>PaperPlain</td>
<td>-1.82%</td>
<td>+0.10 dB</td>
</tr>
<tr>
<td>Pond-Dragonflies</td>
<td>-4.91%</td>
<td>+0.16 dB</td>
</tr>
<tr>
<td>Squirrel</td>
<td>-4.76%</td>
<td>+0.19 dB</td>
</tr>
</tbody>
</table>

![Diagram](Image)

Figure 4: Rate distortion performance of two different coding strategies for the PaintingTilting (1520 × 1080) texture video sequence. A bit saving of 14.0% is achieved by using the discrete cosine-based reference $W_{\text{cosp}}$ in addition to the usual reference $f_x$. Rate (Mbps)

![Diagram](Image)

Figure 5: Prediction unit (PU) structure for frame 5 of the 1080p PaintingTilting texture video sequence [7] produced by the HM encoder.

![Diagram](Image)

Figure 6: Prediction unit (PU) structure for frame 5 of the 1080p PaintingTilting texture video sequence [7] produced by the proposed modification to the HM encoder.