GRAPH-BASED TRANSFORMS FOR INTER PREDICTED VIDEO CODING

Hilmi E. Egilmez†, Amir Said*, Yung-Hsuan Chao† and Antonio Ortega†

†Signal & Image Processing Institute, University of Southern California, Los Angeles, CA, USA
*Qualcomm Technologies, San Diego, CA, USA
hegilmez@usc.edu, asaid@qti.qualcomm.com, yunghsuc@usc.edu, ortega@siip.usc.edu

ABSTRACT
In video coding, motion compensation is an essential tool to obtain residual block signals whose transform coefficients are encoded. This paper proposes novel graph-based transforms (GBTs) for coding inter-predicted residual block signals. Our contribution is twofold: (i) We develop edge adaptive GBTs (EA-GBTs) derived from graphs estimated from residual blocks, and (ii) we design template adaptive GBTs (TA-GBTs) by introducing simplified graph templates generating different set of GBTs with low transform signaling overhead. Our experimental results show that proposed methods significantly outperform traditional DCT and KLT in terms of rate-distortion performance.

Index Terms—Transform, signal processing on graphs, graph-based transforms, video coding, video compression.

1. INTRODUCTION
In video coding standards including HEVC [1], inter-prediction is a very important building block that significantly improves coding efficiency by exploiting high temporal redundancy between video blocks. In general, samples of residual blocks obtained from inter-prediction have low energy, so their transform coefficients can be efficiently encoded. However, some residual blocks may have high energy due to high motion activity and occlusions, so that better energy compacting transforms are needed to improve coding gains. Typically, in conventional video coding architectures as shown in Fig. 1, a fixed transform such as discrete cosine transform (DCT) is employed to accommodate complexity constraints of encoding. The main problem of using a fixed block linear transform is the implicit assumption that all residual blocks have the same isotropic statistical properties. Yet in practice, residual blocks can have very different statistical characteristics depending on video content. Better compression can be achieved by using different transforms that can adapt to statistical properties of residual blocks. But, such adaptation requires to encode additional side information, called transform signaling overhead, that is used by the decoder to identify the transforms used at the encoder. Therefore, it is important to design transforms that adapt common residual block characteristics with low signaling overhead.

This paper presents two different type of transforms exploiting statistical characteristics of inter-predicted residual blocks. The proposed transforms fall into the category of graph-based transforms (GBTs) where we first design a graph capturing some signal characteristics observed from inter-predicted residual blocks, and associated orthogonal transforms are then derived from the designated graph. In our first design, which is edge adaptive GBT (EA-GBT), we allow flexible adaptation for each residual block. Firstly, edge detection is performed for each residual block, and based on detected edges we construct a weighted graph which captures signal variation characteristics in the block. Then, an EA-GBT is generated using the weighted graph. Note that, this method can create a large signaling overhead depending on the graph information has to be sent to the decoder. Our second design proposes template adaptive GBTs (TA-GBTs) which are derived based on a set of simplified graph templates capturing basic statistical characteristics of inter-predicted residual blocks. Thus, graph information can be efficiently sent to the decoder via signaling indexes of corresponding graph templates. By selecting different subsets of graph templates, the signaling overhead can be significantly reduced without losing adaptivity, especially when a few graph templates are sufficient to capture block signal characteristics.

In the literature, several adaptive transform approaches have been proposed. Most similar to our work, Shen et.al. [2] propose edge adaptive transforms (EAT) specifically for depth map compression. Although our paper adopts some basic concepts originally introduced in [2] for designing EA-GBTs, our graph construction method is different. Hu et.al. [3] extends EATs by optimizing weak-link weights for piecewise smooth image compression. In both [2] and [3], authors propose methods specific to depth map compression, but our work focuses on encoding inter-predicted residual blocks. Related to inter-predicted coding, Liu and Flierl [4] propose motion adaptive transforms based on vertex weighted graphs for coding motion-connected pixels. Their approach is not block based and in their graph construction, unlike in our work, vertex weights are adjusted using a measure called motion scale factor. Most of the related recent works are on intra-predicted adaptive transforms. In [5], Takamura and Shimizu develop intra-mode dependent KLTs, and Han et.al. [6] introduce hybrid DCT/ADST transform for intra-
predicted transform coding. To best of our knowledge, our paper is the first work that proposes GBTs for encoding inter-predicted residual blocks by exploiting their statistical characteristics.

The rest of the paper is organized as follows. In Section 2 we introduce GBTs. Section 3 discusses inter-predicted residual signal characteristics used in designing proposed GBTs. In Section 4, the proposed EA-GBTs and TA-GBTs are described. The experimental results are presented in Section 5, and Section 6 draws some conclusions based on experimental results.

2. PRELIMINARIES

In graph signal processing [7, 8], signals are supported on an undirected, weighted and connected graph, \( G(\mathcal{V}, \mathcal{E}, A) \), where signal values are attached to nodes of the graph \( \mathcal{V} \) and its links \( \mathcal{E} \) capture inter-sample relations among signal’s samples. The adjacency matrix, \( A \), represents the graph’s link weights. For a given graph, \( G(\mathcal{V}, \mathcal{E}, A) \), we define graph-based transforms (GBTs) using its combinatorial Laplacian,

\[
L = D - A
\]

where \( D \) is the diagonal degree matrix. In order to find the GBT associated with graph \( G \), we perform eigen-decomposition of the graph Laplacian, that is

\[
L = T \Lambda T^T
\]

where the columns of \( T \) are the basis vectors of the corresponding GBT. Since \( L \) is a real symmetric matrix, it has a complete set of orthonormal eigenvectors.

A graph is completely defined by an adjacency matrix, so we can create different transforms by designing graph-link weights (i.e., \( A \)). For example as shown in Fig. 2, an image block can be represented as a graph so that different connectivity patterns lead to different interpretations in graph transform domain.

3. INTER-PREDICTED RESIDUAL BLOCK SIGNAL CHARACTERISTICS

In this section, we investigate some statistical properties of inter-predicted residual blocks that we consider in our transform designs. In general, inter-predicted residual block signals have small valued (low energy) samples because of high temporal redundancy among video blocks. This is very important for effective compression, since it leads to sparse quantized coefficients which can be encoded efficiently. However, large prediction errors are possible in case of high motion activity and occlusions which lead to large transform coefficients requiring more bits for encoding. Based on our observations on residual block signals obtained using HEVC encoder (HM-14), residual signal samples that are close to boundaries of the blocks have larger values mainly because of occlusions leading to partial mismatches between reference and predicted blocks. Fig. 3 illustrates sample variance values calculated over \( 8 \times 8 \) residual blocks of Harbour and Soccer sequences\(^1\). Note that for both sequences, sample variance (i.e., energy) is larger around the boundaries and corners of the residual blocks.

Moreover, Fig. 4(a) and (b) show similarity graphs trained for \( 8 \times 8 \) inter-predicted residual blocks over Harbour and Soccer video sequences, respectively. As a measure of inter-pixel similarity, partial correlation values are calculated based on the precision matrix, \( J \), where \( J \) is defined as the inverse of the covariance matrix [9], calculated for each video sequence. The weighted graphs demonstrate that the similarity between the pixels near boundaries of a residual block is smaller compared to the pixels around the center of the block.

It is important to note that the statistical characteristics of inter-predicted residuals discussed in this section are not specific to Harbour and Soccer video sequences. According to our experiments, these characteristics are fairly general and applies to different sequences and residual block sizes. These characteristics are exploited in our GBT designs discussed in the next section.

4. PROPOSED GRAPH-BASED TRANSFORMS

4.1. Edge Adaptive GBT (EA-GBT)

In designing edge adaptive graph based transforms (EA-GBT), we first (i) generate a uniformly weighted graph, then (ii) based on differences between pixels (i.e., edges), graph links are pruned or their weights are adjusted (weakened). By doing this, the transforms associated to designed graphs can exploit different block signal characteristics and therefore GBTs provide better representation of residual signals. In particular, we propose to implement following steps to construct EA-GBTs:

\(^1\)We show statistical properties of Harbour and Soccer sequences, since both have high motion activity.
1. Based on the size of the residual block of interest, we create a nearest neighbor (4-connected) graph with link weights all equal to 1 as shown in Fig. 2(a) for $8 \times 8$ blocks.

2. Given a residual block, we apply Prewitt operator to calculate gradient in vertical and horizontal direction.

3. We detect edges based on thresholding on gradient values.

4. Depending on angle value (directionality) of an edge, the weights of some graph links are reduced.

5. Weak graph link weights can be chosen in the range of [0,1]. Based on our experiments, instead of assigning zero weights (may lead to disconnected components), small weights provide better compression. To reduce signaling overhead, we experimentally select a single weak link weight set to 0.001.

6. After designing a graph, the associated GBT is constructed as discussed in Section 2.

Fig. 5 illustrates a sample graph design, obtained by the procedure above, where link weights of the original 4-connected graph is weakened based on the edges observed in a given residual block. Thus, the transforms associated to constructed graphs can adapt to different residual block signals. Although the resulting transforms can provide efficient coding for transform coefficients, the overall coding performance may not be sufficient due to signaling overhead of graph information, especially if multiple weak link weights are used. To address this problem, we propose to use a single weak link weight so that an edge-map codec such as arithmetic edge encoder (AEC) [10] can be employed to efficiently send graph information. In addition, based on our experiments, signaling graph information for small blocks (e.g., $4 \times 4$) may result in excessive bit overheads. In order to efficiently encode graph information for such blocks, we propose to combine the graphs obtained from neighboring blocks and then the combined graph is encoded using the AEC encoder.

4.2. Template Adaptive GBT (TA-GBT)

In this section, we propose a fixed set of GBTs derived from a set of graph templates considering the inter-predicted residual signal characteristics discussed in Section 3. The main observation we exploit in our design is that sharp transitions (i.e., most of the energy) appear around the corners of inter-predicted residual blocks. This is mainly due to mismatched regions (i.e., occlusions) in inter-prediction. The basic building blocks of the proposed graph template construction are as follows:

1. We choose a base graph that is a uniformly weighted graph, $G_{uni}$, where two examples are shown in Fig. 2. In this work, we employ nearest-neighbor image model, so 4-connected grid graph is used (see Fig. 2(a)).

2. By adjusting a subset of links’ weights in $G_{uni}$, $K$ different graphs are constructed. These different graphs are called graph templates $\{G_k\}_{k=1}^K$ which define GBTs.

3. The statistical properties of inter-predicted residual blocks can be captured by reducing the weights of links in $G_{uni}$ connecting pixels around the corners of a transform block. Particularly in this work, $K=16$ templates are generated by repeating different combinations of a rectangular pattern to denote weak links around the corners of the graph, $G_{uni}$.

4. For a selected set of graph templates, the associated GBTs are constructed as discussed in Section 2.

Fig. 6 shows five of the sixteen graph templates designed for $8 \times 8$ transform blocks which lead to 16 different GBTs. Similarly, we also generate 16 transforms for $4 \times 4$ residual blocks. Note that the first template corresponds to traditional 2-D DCT [9].

In order to adaptively select the best transform, we introduce a graph Laplacian based quadratic cost which measures residual signal variation on a given graph. Formally, for a given residual block signal $d$ we select the transform whose associated graph representation $(G_k)$ solves the following optimization problem,

$$\text{minimize } \sum_{i=1}^N \lambda_i a_i^2$$

where $L(G_k)$ is the combinatorial graph Laplacian of graph $G_k$, $a$ is the vector of transform coefficients, $N$ is the number of samples in the residual block, $\lambda_i$ denotes the eigenvalues of the graph Laplacian in increasing order (i.e., $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_{N-1}$) and $a_i$ is the transform coefficient associated with $\lambda_i$. This criterion is a way of measuring energy compaction, so that the larger $\lambda$ is, the larger penalty for its transform coefficients are. Since the first eigenvalue, $\lambda_1$, is zero [7], then

$$\sum_{i=1}^N \lambda_i a_i^2 = \lambda_2 \sum_{i=2}^N \lambda_i a_i^2.$$  

which induces no penalty for transform coefficient $a_1$ (i.e., DC component).

5. RESULTS

In this section, we compare the rate-distortion (RD) performance of the proposed transforms by benchmarking against DCT and KLT. In our simulations, we generate residual block signals for five test sequences, Foreman, Mobile, City, Harbour and Soccer, using HEVC (HM-14) encoder where transform units are fixed to either $4 \times 4$ or $8 \times 8$. We test the performance of different transforms on inter-predicted residual blocks only. After transforming residual blocks, the transform coefficients are uniformly quantized and then encoded using a symbol grouping-based arithmetic entropy encoder called AGP, which uses an amplitude group partition technique to efficiently encode image transform coefficients [11]. The AGP encoder allows us to fairly compare the rate-distortion performance of different transforms since AGP can flexibly learn and exploit amplitude


Table 1: Percentage reduction in bitrate (bits/pixel) with respect to average bitrate obtained using DCT.

<table>
<thead>
<tr>
<th>PSNR (dB)</th>
<th>Transform</th>
<th>4 × 4 block transform</th>
<th>8 × 8 block transform</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Foreman</td>
<td>Mobile</td>
<td>City</td>
</tr>
<tr>
<td>32</td>
<td>EA-GBT(RO)</td>
<td>-90.44</td>
<td>18.40</td>
<td>3.98</td>
</tr>
<tr>
<td></td>
<td>EA-GBT</td>
<td>-423.02</td>
<td>-21.46</td>
<td>-99.45</td>
</tr>
<tr>
<td></td>
<td>TA-GBT</td>
<td>0.77</td>
<td>11.76</td>
<td>6.38</td>
</tr>
<tr>
<td>34</td>
<td>EA-GBT(RO)</td>
<td>7.19</td>
<td>15.13</td>
<td>18.43</td>
</tr>
<tr>
<td></td>
<td>TA-GBT</td>
<td>7.61</td>
<td>8.88</td>
<td>5.83</td>
</tr>
<tr>
<td></td>
<td>KLT</td>
<td>-0.20</td>
<td>1.49</td>
<td>5.38</td>
</tr>
<tr>
<td>36</td>
<td>EA-GBT(RO)</td>
<td>20.44</td>
<td>10.76</td>
<td>13.79</td>
</tr>
<tr>
<td></td>
<td>TA-GBT</td>
<td>7.06</td>
<td>6.76</td>
<td>4.58</td>
</tr>
<tr>
<td></td>
<td>KLT</td>
<td>2.01</td>
<td>0.58</td>
<td>3.85</td>
</tr>
</tbody>
</table>

Fig. 7: Average PSNR vs. BPP results (left) for 4 × 4 blocks and (right) for 8 × 8 blocks. EA-GBT(RO) corresponds to the method (only applied to 4 × 4 blocks) that reduces signaling overhead of EA-GBT by combining graph information at neighboring blocks.

distribution of transform coefficients. For ordering of the quantized coefficients, we employ zig-zag scanning for DCT coefficients, and descending and ascending order of eigenvalues are used for KLT and GBT’s coefficients, respectively. To send transform signaling information for EA-GBT, we use the arithmetic edge codec (AEC) [10] to efficiently code graph information. To further reduce the overhead of graph coding for 4 × 4 blocks (EA-GBT(RO)), we combine the graphs obtained from neighboring blocks and the resulting larger graph is encoded using AEC. For TA-GBT, the transform indexes are signaled as the side information. After decoding the quantized transform coefficients using AGP decoder, we reconstruct the video blocks and measure PSNR with respect to the original video blocks.

The average RD performances of different transforms are presented in Fig. 7 in terms of PSNR and total bits spent per-pixel (BPP) for encoding quantized transform coefficients, motion vectors and transform signaling overheads. More comprehensive results are available in Table 1 where we show percent bit reductions for each video sequence gained by using GBTs and KLT at different PSNR values (i.e., 32, 34 and 36 dBs) with respect to using DCT. Average percent reductions (corresponding to Fig. 7) are also given in Table 1. Note that, positive values in the table means that the better RD performance is achieved compared to using DCT. According to these results:

- For 4 × 4 blocks, RD performance of EA-GBT is the worst among all transforms due to the excessive graph signaling overhead. However, the signaling overhead of EA-GBT is significantly reduced by combining the graph information of neighboring blocks (see EA-GBT(RO) in Table 1 and in Fig. 7).
- EA-GBT(RO) and EA-GBT outperform all other transforms at high-rate coding of 4 × 4 and 8 × 8 blocks, respectively. On the other hand, TA-GBT provides a reasonable coding gain for both low-rate and high-rate coding with respect to DCT.

6. CONCLUSIONS

In this paper, we have proposed two novel transforms, EA-GBT and TA-GBT, for inter-predicted residual block signals, and their rate-distortion (RD) performance is compared against traditional DCT and KLT. The inspection of the experimental results lead us to following conclusions:

- Proposed EA-GBT provides 9.9% coding gain at 34dB PSNR with respect to DCT for 8 × 8 residual blocks. For 4 × 4 blocks, 15.2% gain can be achieved using EA-GBT. However, at low bitrates corresponding to 30-32dB PSNR, the graph signaling overhead exceeds the bit reduction gained using EA-GBT. Therefore, we propose to use TA-GBT for coding at low bitrates.
- Proposed TA-GBT nicely captures the characteristics of 4 × 4 residual blocks with low transform signaling overhead. At 34dB PSNR, it provides 6.2% bitrate reduction with respect to DCT on average. For 8 × 8 blocks, the reduction is relatively less, that is 4.2%. Using more graph templates can improve the coding gain, since more different signal characteristics can be captured in 8 × 8 or larger blocks.
- For 4 × 4 blocks, it is inefficient to directly send graphs as the side-information. By exploiting the graph information from the neighboring blocks, we show that the signaling overhead can be significantly reduced.
7. REFERENCES


