ADAPTIVE COMPRESSED SENSING FOR DEPTHMAP COMPRESSION USING GRAPH-BASED TRANSFORM
Sungwon Lee and Antonio Ortega
Signal and Image Processing Institute, University of Southern California, Los Angeles, CA, USA

Dephmap Compression for Multiview Video
- Precise edge information crucial in depth image based rendering (DIBR) system
- Traditional DCT (H.264/AVC) not efficient for arbitrary edges
- Requires edge-adaptive transform

Related Work
- Graph-based transform (GBT) [3]
- GBT for Depthmap Compression
- CS-based depthmap compression [4, 5]

GBT for Depthmap Compression
- Efficiently sparsifies arbitrary edges in depthmap [6]
- Construction procedure:
  - Given graph, $G(V, E)$, construct adjacency matrix, $A$
  - Compute Laplacian matrix, $L$
  - For a graph, $G(V, E)$, construct adjacency matrix, $A$

Average Mutual Coherence, $\mu_{avg}$
- Can directly compute mutual coherence, $\mu$, between Hadamard and GBT [7]
- However, complexity significantly increases because
- Expensive GBT construction due to EVD of $L$
- Need to compute $\mu_{avg}$ in every iteration of the algorithm
- Propose $\mu_{avg}$ as an approximation of $\mu$
  - $\mu_{avg} = \frac{\mu_{max}}{\mu_{min} \sqrt{N_g}}$
  - $\mu_{max}$ depends on the maximum size of group
  - $\mu_{min}$ depends on Hadamard matrix ($\Phi$) and edge map (E)
  - $\mu_{avg}$ can compute without GBT construction

Compressed Sensing (CS) for Depthmap Compression
- Sensing basis, $\Phi$, Hadamard matrix for simplicity
- Sparsifying basis, $\Psi$: GBT for sparser representation of depthmap

Simulation Result
- Encode 4 Hadamard measurements for CS
- Extra bits for adjacency matrix, $A$
- Choose the best coding method by RD optimization
- Four QP values: 24, 28, 32, and 36

Table: Performance comparison of different methods
<table>
<thead>
<tr>
<th>Method</th>
<th>BD-PSNR</th>
<th>BD-Bitrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.264/AVC</td>
<td>-8 dB</td>
<td>1.2 kbps</td>
</tr>
<tr>
<td>GBT (optimal A)</td>
<td>-3.9 dB</td>
<td>1.2 kbps</td>
</tr>
<tr>
<td>CS+GBT (optimal A)</td>
<td>-3.9 dB</td>
<td>1.2 kbps</td>
</tr>
<tr>
<td>CS+GBT (optimal CS-A)</td>
<td>-3.9 dB</td>
<td>1.2 kbps</td>
</tr>
</tbody>
</table>

Conclusion
- Propose a block-based CS framework applicable to depthmap compression
- Find an optimal edgemap for CS with approximate average mutual coherence without GBT
- Possible extension of CS with GBT for other types of data

References