A Compression Technique for Piecewise Smooth Images based on Transform Coding

Lince Varghese Student [Signal Processing] College of Engineering,Poonjar

ABSTRACT

Image compression has a wide range of application since it leads to reduction in storage space and easy transmission. Piecewise smooth image consists of sharp edge boundaries and smooth interior surfaces. This paper deals with compression of Piecewise smooth images using Graph Fourier Transform and Discrete Cosine Transform. In order to obtain better quality of reconstructed image blocks contains edge boundaries are transformed using DCT and smooth regions are transformed using both weighted GFT and unweighted GFT. In order to reduce the computational complexity, low pass filter and down sample a high resolution pixel block to obtain a low resolution one at the encoder, so that LR-GFT can be employed. At the decoder upsampling and interpolation are performed so that sharp edge boundaries can be preserved.

Keywords

Image compression, Graph Fourier Transform, Discrete Cosine Transform, Piecewise smooth images

1. INTRODUCTION

Most of the information consists of different types of redundancies. Removal of these redundancies will not affect the quality of the original message. Image compression is a process of reducing the amount of data required to represent a given quantity of information without any loss of its contents. Most images contain information that is ignored by the human visual system. Since increase in image related applications have created an issue of image storing and transmission, data compression it is one of the most used techniques in the field of image processing.Image compression mainly consists of three steps. At the encoder side divide the input image into different blocks, apply different transformations and finally entropy coding on the coefficients. All inverse operations are performed at the decoder side. Depending upon distinct transformations and entropy coding techniques, there are so many compression techniques are available.

In transform coding, initially DCT was most commonly used image compression technique. DCT provides simplicity and good performance in compression. As it is applied on blocked image correlation across the block boundaries cannot be eliminated. It leads to blocking artifacts specifically at low bit rate. This drawback was overcome by wavelet transform [2]. Image compression using wavelet transform is popular as it gives time domain and frequency domain analysis of data. Wavelet transform can be directly applied to whole image without blocking it or it can be applies after dividing the input image into blocks. Wavelet based coding is more robust under transmission and decoding errors [3]. Recent trend is to use hybrid technique for image compression. In hybrid image compression technique, one transform is combined with another transform to incorporate the advantages of both transforms. In this work Graph Fourier Transform and

Shine P. James Assistant Professor College of Engineering,Poonjar

Wavelet Transform are used for the compression of Piecewise smooth images.



Fig 1: Examples of PWS images: (a) a depth map of an image sequence Teddy; (b) a computer-graphics image Dude.

2. MULTIRESOLUTION GRAPH FOURIER TRANSFORM SYSTEM

This section provides an overview of proposed coding system for compression of PWS images as shown in Figure 2. For a given a PWS image, the encoding and decoding procedures are as follows.





2.1 Encoder

At the encoder, first we will detect prominent boundaries (large inter-pixel intensity difference) in the HR image via hard thresholding of image gradients using sobel operator. Here the threshold is set based on the mean and variance of the input image, so that the boundary detection is adaptive to the image statistics. Then we will encode HR boundaries losslessly using Huffman coding for H.264 intra prediction and interpolation at the decoder. Then apply Huffman coding for the entire image, which avoids initialization for each block and efficiently encodes long continuous boundaries in the image. Then for each $K\sqrt{N} \times K\sqrt{N}$ target pixel block considering a down sampling factor K = 2 and we execute the following three steps.

First, this paper performs intra prediction. H.264 intra prediction mode efficiently minimizes the prediction error by predicting within the limit of detected HR boundaries, thus reducing bits required for coding of the residual signal. Second, this method try two types of transforms for transform coding of the residual block. It performs, i) fixed DCT on the original HR residual block (HR-DCT) and ii) a pre-computed LR GFT (LR-GFT) on the down-sampled LR residual block. Then choose best transform for each block which gives minimum error. Before transform coding using LR-GFT, however, first adaptively low-pass-filter and down-sample the

 $K\sqrt{N} \times K\sqrt{N}$ block uniformly. Low-pass filtering is first used to avoid the effects of aliasing caused by downsampling. An edge-adaptive low-pass filtering is performed in the pixel domain in order to preserve of sharp boundaries. Specifically, a pixel is low-pass-filtered by taking average of its neighboring pixels on HR boundaries within a $(2K-1) \times (2K-1)$ window centering at the pixel which is to-be-filtered. The main advantage of this edge-adaptive lowpass filtering is that filtering across arbitrary-shape boundaries will not exist, so pixels across boundaries will not corrupt each other through filtering. For the employment of the HR-DCT and LR-GFT pre-compute the optimal transforms and store them in a lookup table a priori. During coding, try each of the transforms and choose the one which gives the better performance. In frequency domain two types of transforms, HR-DCT and LR-GFT are employed to adapt to various block characteristics.

HR-DCT is sufficient for blocks where edge-adaptive lowpass filtering would result in significant amount of energy loss. If energy loss is negligible during low pass filtering, LR-GFT would result in a larger coding gain. If a given block is smooth then the LR-GFT will default to the DCT in LR, and it would give a larger gain than HR-DCT due to downsampling.

After optimal transform is chosen from the two transform candidates, quantize and entropy-encode the resulting transform coefficients for further transmission to the decoder. The transform index recognizing the selected transform is also encoded, so that suitable inverse transform can be achieved at the decoder.

2.2 Decoder

At the decoder, first perform inverse quantization and inverse transform for the reconstruction of the residual block. The transform index is used to distinguish the transform chosen at the encoder for transform coding. Secondly, if LR-GFT is employed, up-sample the reconstructed $K\sqrt{N} \times K\sqrt{N}$ LR residual block to the original resolution $K\sqrt{N} \times K\sqrt{N}$, and then fill in missing pixels via image-based edge-adaptive interpolation, where a pixel x is interpolated by taking average of its neighboring pixels on the same side of boundaries within a $(2K-1) \times (2K-1)$ window centering at pixel x. Finally, the $K\sqrt{N} \times K\sqrt{N}$ block is reconstructed by adding the intra predictor to the residual block.

3. GRAPH FOURIER TRANSFORM FOR IMAGES

A graph $G = \{V, E, W\}$ consists of a finite set of vertices V with cardinality of N, a set of edges E connecting vertices, and a weighted adjacency matrix W. W Is a real $N \times N$ matrix, where $W_{i,j}$ is the weight assigned to the edge (i, j) connecting vertices i and j. Here we are considering only undirected graphs, which correspond to symmetric weighted adjacency matrices, i.e., $W_{i,j} = W_{j,i}$. We also assume weights are non-negative, i.e., $W_{i,j} \ge 0$.



Fig 3: An example of constructing GFT from a 2x2 pixel block[1]

Since we have different variants of Laplacian matrices, here we are using unnormalized combinatorial graph Laplacian, which is defined as L = D - W, where D is the degree matrix - a diagonal matrix whose *i* th diagonal element is the sum of all elements in the *i* th row of W, i.e., $D_{i,j} = \sum_{j=1}^{N} W_{i,j}$. Since Laplacian matrix is a real symmetric matrix, it adopts a set of real eigenvalues $\{\lambda_l\}_{l=0,1,...,N-1}$ with a complete set of orthonormal eigenvectors $\zeta \psi_l = \lambda_l \psi_l$, for l=0,1,...,N-1. We employ this Laplacian matrix mainly for two reasons.

First, because elements in each row of ζ add to zero by construction, 0 is guaranteed to be an eigenvalue with $[1, \dots, 1]^T$ as the corresponding eigenvector. This means a frequency domain interpretation of GFT, where the eigenvalues ls are the graph frequencies, which have a DC component, which is beneficial for the compression of PWS images where most of the regions are smooth.

Second, GFT defaults to the well known DCT when defined for a line graph (corresponding to the 1D DCT) or a 4connectivity graph (2D DCT) with all edge weights equal to 1 [27]. That means GFT is at least as good as the DCT in sparse signal depiction if the weights are chosen in this way. Because of the above two desirable properties, the unnormalized Laplacian matrix is used in this work.

The graph Laplacian can be used to describe the total variation of the signal with respect to the graph; i.e., for any

signal $x \in \mathbb{R}^N$ residing on the vertices of a graph with the graph Laplacian ζ , we can write

$$x^{T}\zeta x = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} W_{i,j} (x_{i} - x_{j})^{2}$$
(1)

 $x^{T} \zeta x$ is small when x has similar values at each pair of vertices *i* and *j* connected by an edge, or when the weight $W_{i,j}$ is small for an edge connecting *i* and *j* with dissimilar values. Thus, a signal is smooth (mostly low-frequency components) with respect to a graph if the edge weights capture the similarity of connected pixels in the signal. Since

 $x^{T} \zeta x$ is small if a signal is smooth on the graph (thus a sparse representation in the GFT domain).

The eigenvectors $\{\psi_l\}_{l=0,1,...,N-1}$ of the Laplacian matrix are then used to define the GFT. Formally, for any signal $x \in \mathbb{R}^N$ residing on the vertices of *G*, its GFT $\hat{x} \in \mathbb{R}^N$ is defined as

$$\hat{x}(l) = \left\langle \psi_l, x \right\rangle = \sum_{n=1}^{N} \bar{\psi}_l(n) x(n), l = 0, 1, ..., N - 1.$$
(2)

The inverse GFT follows as

$$x(n) = \sum_{l=0}^{N-1} \hat{x}(l) \psi_l(n), n = 1, 2, \dots, N$$
(3)

Having defined the GFT, we an example of how to construct a GFT from a given unweighted graph, defined for a pixel block is given below. Given a $\sqrt{N} \times \sqrt{N}$ pixel block, first treat each pixel in the block as a vertex in a graph *G* and connect it to its four immediate neighbors, resulting in a 4-connectivity graph which is shown in Figure 3 for an illustration. Second, if there is a large discrepancy in values between two neighboring pixels, eliminate their connection. Given the connectivity graph, then define the adjacency matrix W, where $W_{i,j} = W_{j,i} = 1$ if pixel positions *i* and *j* are connected, and 0 otherwise. The degree matrix D can then

be computed. In the third step, using computed W and D, compute the graph Laplacian matrix L = D - W. Then stack

pixels in the $\sqrt{N} \times \sqrt{N}$ patch into a length-N vector x and compute the GFT according to (2).

4. EXAMPLES FOR OPTIMAL GFTS FOR DIFFERENT BLOCKS

Figure 4 shows the two flavors of GFTs to capture various correlations in pixel blocks. Figure 4(a) shows an example for block containing pixel pairs which having both strong and zero correlations. UGFT is selected during mode decision based on RD costs, where the given block is divided into two separate partitions and a connected graph is obtained in each partition. The resulting transform representation is one DC coefficient for the description of each partition, as shown at the bottom of Figure 4(a).



Figure 4: An illustration of UGFT and WGFT. (a) An example block containing pixel pairs with strong and zero correlations, and its signal representation in the UGFT domain. (b) An example block containing pixel pairs with strong and weak correlations, and its signal representation in the WGFT domain.

Figure 4(b) shows an example for block containing pixel pairs with strong and weak correlations. As a result WGFT is chosen from mode decision, where a graph containing edge weights c (here c = 0.13) is constructed on the block. The resulting transform coefficients consist of one large DC term and one small AC term, shown at the bottom of Figure 4(b).





Next, is an illustrative example of optimal UGFTs for the same block but at different target bit rates. Given a pixel block with three smooth regions as shown in Figure 5, the minimal representation cost at high bit rates is achieved when it is divided into three connected components corresponding to the three smooth regions. This is because this obtained UGFT results in only three DC coefficients, which leads to the minimal and significantly smaller cost of quantized transform coefficients (compared to other UGFTs which yields in many high frequency coefficients) at high bit rates, which together with the description cost results in the minimal representation cost. At medium rates, the larger quantization parameter (QP) quantizes more coefficients to set to zero, and a simpler UGFT with similar quantized transform coefficient cost but

smaller transform description cost would be more attractive, resulting in the graph in Figure 5(b). Finally, at low rates, the large QP quantizes most coefficients to zero, and the simplest UGFT is best, as shown in Figure 5(c).

5. DEFINITION OF SEARCH SPACE FOR GFTS

Here considering pixels that are neighbors in the 4connectivity graph as random variables, and consider a discrete set of probable weights to be chosen for each edge connecting two pixels. These weights correspond to three representative classes of the correlation between two pixels: 1) strong correlation between the pixels, which models pixel pairs in smooth regions occupying of the foreground/background; 2) zero correlation between the two pixels, occurring when they bestride sharp boundaries between the foreground and background; and 3) weak correlation between the pixels that correspond to distinctly different parts of the same foreground/background object. See Figure 6 for an illustration. Correspondingly, the weights are assigned to be C = f1, 0, cg, where c is a carefully chosen real value between 0 and 1. Having defined the edge weight set, for ease of computation we further divide the optimization problem into two sub-problems with two corresponding nonoverlapping GFT search spaces:

1. Unweighted GFT (UGFT), with C = f1, 0g for blocks that can be well described by pixel pairs with strong and zero correlations only; and

2. Weighted GFT (WGFT), with C = f1, cg for blocks that can be well described by pixel pairs with strong and weak correlations only.



Figure 6: An intuitive illustration of how different transforms adapt to pixel blocks with different pixel correlations in the depth map of Ballet

If a block can be described by pixel pairs having strong correlation only (i.e., a smooth block), then the GFT defaults to the DCT. See Figure 3.5 for an illustration. In a nutshell, the WGFT cogitates only graphs that describe a single connected component (i.e., only one DC coefficient in any WGFT representation of the signal). The UGFT considers graphs of numerous connected components, where each component is connected by edges and having edge weight as 1. A connected component with a pair of disconnected neighboring pixels occurs rarely in practice, and thus is not considered in the UGFT optimization. For a given input coding block we will search both the transform space for the best possible signal representation.

6. FAST IMPLEMENTATION OF GFTS

This section consists of a method for fast implementation of GFT for practical deployment. The online Eigen decomposition for the construction of GFT is a problem to real time implementation, which can be avoided by precalculating and storing most suitabe GFTs in a table for simple lookups. First detail the construction of GFT lookup table, and then we compare the complexity of table lookup against that of online eigen-decomposition.

The table size could be large if we simply store all the obtained GFTs. A large table would always lead to high requirement in storage space and expensive lookup. Hence, it would be better to construct a GFT lookup table having relatively small size. Specifically, the lookup table creation has two strategies: 1) Perform GFT on a LR block (4 - 4), which admits a smaller space of GFT variants to begin with. 2) Only the most frequently used LR-GFTs are stored.

Due to self-similarities, the same or similar structures are likely to persist throughout. Hence, the underlying LR-GFTs with respect to those structures are frequently used. Thus store only the most frequently used LR-GFTs in lookup table, while covering a large fraction of the total used LR-GFTs. Having constructed the lookup table, during encoding search for the GFT (including UGFT and WGFT) for a given block. For each block, search for the optimum GFT during table look-up. Then transmit the table index losslessly to indicate which LR-GFT is employed for the given block, so that the decoder is able to identify the correct inverse transform.

7. EXPERIMENTAL RESULTS

For the performance evaluation of the work, a Piecewise smooth image Teddy with dimension 450x375 is taken as the input image. The input image and the reconstructed images are shown in Figure 7.



Figure7: (a) Input image Teddy (b) Reconstructed image Teddy (c) Input image Cones (b) Reconstructed image Cones.

7.1 Performance Evaluation

In order to obtain the performance evaluation the PSNR values at different bits per pixel are obtained and the corresponding graph is given in Figure 8.





Fig 8: Performance evaluation for the image teddy

8. CONCLUSION

Image compression has a great importance since it reduces storage space and makes the transmission of data easier. This work produces efficient compression of Piecewise Smooth Images using both Graph Fourier Transform and Discrete Cosine Transform. The compressed data and the reconstructed images are obtained and Peak Signal to Noise Ratio at different bits per pixel is tabulated successfully. Further, here we employ two techniques for practical implementation of GFT. One is the MR scheme where GFT is applied over a low pass filtered and down-sampled version of a high-resolution block. The next is the pre-computation of the most popular GFTs in a stored table for simple lookup instead of real time eigen-decomposition. It is possible to implement this method as a coding mode during compression of general images, so that when a code block is deemed PWS, this coding scheme can be deployed.

9. REFERENCES

- Wei hu, Gene Cheung, Antonio Ortega, Oscar C. Au "Multiresolution Graph Fourier Transform for compression of piecewise smooth images", IEEE transactions on image processing, vol. 24, no. 1, January 2015
- [2] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2010.
- [3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [4] J. Huang and P. M. Schultheiss, "Block quantization of correlated Gaussian random variables," *IEEE Trans. Commun. Syst.*, vol. 11, no. 3, pp. 289–296, Sep. 1963.
- [5] V. K. Goyal, J. Zhuang, and M. Vetterli, "Transform coding with backward adaptive updates," *IEEE Trans. Inf. Theory*, vol. 46, no. 4,pp. 1623–1633, Jul. 2000.
- [6] S. Jana and P. Moulin, "Optimality of KLT for high-rate transform coding of Gaussian vector-scale mixtures: Application to reconstruction, estimation, and classification," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 4049–4067, Sep. 2006.

- [7] M. Effros, H. Feng, and K. Zeger, "Suboptimality of the Karhunen–Loeve transform for transform coding," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1605–1619, Aug. 2004.
- [8] A. K. Jain, Fundamentals of Digital Image Processing. Upper Saddle River, NJ, USA: Prentice-Hall, 1989.
- [9] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 598–603, Jul. 2003.
- [10] V. K. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 9–21, Sep. 2001.
- [11] P. Wan, Y. Feng, G. Cheung, I. V. Bajic, O. C. Au, and Y. Ji, "3D motion in visual saliency modeling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.* (*ICASSP*), Vancouver, BC, Canada, May 2013, pp. 1831–1835.
- [12] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," *Proc. SPIE*, vol. 7443, p. 74430T, Feb. 2009.
- [13] S.-C. Han and C. I. Podilchuk, "Video compression with dense motion fields," *IEEE Trans. Image Process.*, vol. 10, no. 11, pp. 1605–1612, Nov. 2001.
- [14] D. S. Hochbaum, "An efficient and effective tool for image segmentation, total variations and regularization," in *Proc. 3rd Int. Conf. Scale Space Variational Methods Comput. Vis.*, vol. 6667, 2011, pp. 338–349.
- [15] J. Shi and J. Malik, "Normalized cuts and image segmentation,"*IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
- [16] R. Shukla, P. L. Dragotti, M. N. Do, and M. Vetterli, "Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 343–359, Mar. 2005.
- [17] Y. Morvan, P. H. N. de With, and D. Farin, "Plateletbased coding of depth maps for the transmission of multiview images," *Proc. SPIE*, vol. 6055, Jan. 2006, p. 60550K.
- [18] P. Merkle *et al.*, "The effects of multiview depth video compression on multiview rendering," *Signal Process.*, *Image Commun.*, vol. 24, nos. 1–2, pp. 73–88, Jan. 2009.
- [19] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Sep. 2012, pp. 1541–1544.
- [20] I. Daribo, D. Florencio, and G. Cheung, "Arbitrarily shaped motion prediction for depth video compression using arithmetic edge coding," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4696–4708, Nov. 2014.
- [21] M. B. Wakin, J. K. Romberg, H. Choi, and R. G. Baraniuk, "Wavelet-domain approximation and compression of 2002.