# POLITECNICO DI TORINO
## Repository ISTITUZIONALE

## Graph transform learning for image compression

(Article begins on next page)

15 May 2024

# Graph Transform Learning for Image Compression

Giulia Fracastoro
Department of Electronics and Telecommunications
Politecnico di Torino

Dorina Thanou, Pascal Frossard
Signal Processing Laboratory (LTS4)
École Polytechnique Fédérale de Lausanne

*Abstract*—**In this paper, we propose a new graph-based compression scheme for image coding. Our approach relies on the careful design of a graph that optimizes the overall rate-distortion performance. In particular, we model the pixels as nodes of a graph and we treat the pixel intensities as a signal living on an unknown graph topology. We then introduce a novel graph learning algorithm targeted for image compression that uncovers the connectivities between the pixels, by taking into consideration the coding of the image signal and the graph topology in rate-distortion terms. The cost of the graph description is introduced in the optimization problem by treating the edge weights as another graph signal that lies on the dual graph, and minimizing the sparsity of its graph Fourier coefficients (GFT). In this way, we obtain a convex optimization problem whose solution defines the transform of the image signal. The experimental results show that the proposed method outperforms classical fixed transforms such as DCT, and confirm the potential of graph-based methods for adaptive image coding solutions.**

## I. INTRODUCTION

One of the major applications of the emerging field of graph signal processing [1] is image representation and compression. An image can be represented by a graph, where the nodes are the image pixels and the edge weights capture the correlation between adjacent pixels. Such a flexible representation permits to go beyond traditional transform coding by moving from classical fixed transforms such as the discrete cosine transform (DCT) to graph-based transforms that are adapted to the actual signal structure, such as the graph Fourier transform (GFT). By using such transforms, it is possible to obtain a more compact representation of an image, as the energy of the image signal is concentrated in the lowest frequencies. This provides a strong advantage compared to the classical DCT transform especially when the image contains arbitrarily shaped discontinuities. In this case, the DCT transform coefficients in this case are not necessarily sparse and contain many high frequency coefficients with high energy.

However, one of the biggest challenges in graph-based image compression is the choice of the graph and the corresponding transform. A good graph for effective transform coding should lead to easily compressible signal coefficients, at the cost of a small overhead for coding the graph. Most state-of-the-art methods design the graph by considering pairwise similarities among the pixel intensities and they provide significant gain in the coding of piecewise-smooth images [2] [3] [4]. They however report unsatisfactory results on natural images, where the cost required to describe the graph outweighs the coding gain provided by the adaptive graph transform. Since the definition of the graph is often not

straightforward, the problem of designing a graph transform stays critical and may actually represent the major obstacle towards effective compression of images.

In this work, we propose a novel graph-based framework for effective coding of images that takes into account the coding of the images as well as the cost of transmitting the graph. In particular, we introduce an innovative way for coding the graph by treating its edge weights as a graph signal that lies on the dual graph. We then compute the graph Fourier transform of this signal and code its quantized transform coefficients. The choice of the graph is posed as a rate-distortion optimization problem that is cast as a graph learning problem. The cost of coding the image signal is captured by minimizing the smoothness of the image on the learned graph while the transmission cost of the topology is controlled by penalizing the sparsity of the graph Fourier coefficients of the edge weight signal that lies on the dual graph. The solution of our optimization problem is a graph that provides an effective tradeoff between the quality of the transform and its transmission cost. Experimental results on natural images confirm that the proposed algorithm can efficiently infer meaningful graph topologies, which eventually lead to improved coding results compared to non-adaptive methods based on DCT.

A few attempts have been recently proposed to learn the structure and in particular a graph from data observations. In [5], the authors formulate the graph learning problem as a precision matrix estimation with generalized Laplacian constraints. In [6], a sparse combinatorial Laplacian matrix is estimated from the data samples under a smoothness prior. In [7], a new class of transforms called graph template transform is proposed; the authors use a graph template to impose a sparsity pattern and approximate the empirical inverse covariance based on that template. Even if all these methods contain some constraints on the sparsity of the graph, none of them takes into account the real cost of representing, and thus coding, the graph. Instead, in this paper, we go beyond prior art and we fill this gap by defining a new graph learning problem for image compression that takes into account the graph coding cost.

## II. BASIC DEFINITIONS ON GRAPHS

For any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ and $\mathcal{E}$ represent respectively the node and edge sets and $|\mathcal{V}| = N$ and $|\mathcal{E}| = M$, we define the weighted adjacency matrix $W \in \mathbb{R}^{N \times N}$ where $W_{ij}$ is the weight associated to the edge $(i, j)$ connecting

nodes $i$ and $j$. In this paper, we consider undirected graphs with no self loops, thus $W$ is symmetric and has a null diagonal. The graph Laplacian is defined as $L = D - W$, where $D$ is a diagonal matrix whose $i$-th diagonal element $D_{ii}$ is the sum of the weights of all the edges incident to node $i$. Since $L$ is a real symmetric matrix, it is diagonalizable by an orthogonal matrix

$$L = \Psi \Lambda \Psi^T,$$

where $\Psi \in \mathbb{R}^{N \times N}$ is the eigenvector matrix of $L$ that contains the eigenvectors as columns and $\Lambda \in \mathbb{R}^{N \times N}$ is the diagonal eigenvalue matrix where the eigenvalues are sorted in increasing order.

The graph Laplacian $L$ can also be defined using the incidence matrix $B \in \mathbb{R}^{N \times M}$ [8] such that

$$B_{ie} = \begin{cases} 1, & \text{if } e = (i,j) \\ -1, & \text{if } e = (j,i) \\ 0, & \text{otherwise,} \end{cases}$$

where an orientation is chosen arbitrarily for each edge. Then, let $\widehat{W} \in \mathbb{R}^{M \times M}$ be a diagonal matrix where $\widehat{W}_{ee} = W_{ij}$ if $e = (i,j)$. We can define the graph Laplacian $L$ as

$$L = B\widehat{W}B^T. \tag{1}$$

It is important to underline that the graph Laplacian obtained using (1) is independent from the edge orientation in $\mathcal{G}$.

A graph signal $x \in \mathbb{R}^N$ in the vertex domain is a real-valued function defined on the nodes of the graph $\mathcal{G}$, such that $x_i$, $i = 1, \ldots, N$ is the value of the signal at node $i \in \mathcal{V}$ [1]. For an image signal, we can consider an associated graph where the nodes of the graph are the pixels of the image. Then, the smoothness of $x$ on $\mathcal{G}$ can be measured using the Laplacian $L$ [9]

$$x^T L x = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} W_{ij}(x_i - x_j)^2. \tag{2}$$

Eq. (2) shows that a graph signal $x$ is considered to be smooth if strongly connected nodes have similar signal values. Finally, the eigenvectors of the Laplacian are used to define the graph Fourier transform (GFT) [1] of the signal $x$ as follows:

$$\hat{x} = \Psi^T x.$$

The graph signal $x$ can be easily retrieved from $\hat{x}$ by inversion, namely $x = \Psi \hat{x}$. Analogously to the Fourier transform in the Euclidean domain, the GFT is used to describe the graph signal in the Fourier domain.

## III. GRAPH-TRANSFORM OPTIMIZATION

### A. Rate-distortion tradeoff

Graph-based image compression methods use a graph representation of the image signal through its GFT, in order to obtain a data-adaptive transform which captures the main characteristics of the image. The GFT coefficients are then encoded, instead of the signal values. In general, a signal that is smooth on a graph has its energy concentrated in

the low frequency coefficients of the GFT, hence it is easily compressible. To obtain good compression performance, the graph should therefore be chosen such that it leads to a smooth representation of the signal. On the other hand, it should also be easy to encode, since it has to be transmitted to the decoder for signal reconstruction. Often, the cost of the graph representation outweighs the benefits of using an adaptive transform for signal representation. In order to find a good balance between graph signal representation benefits and coding costs, we introduce a new graph learning approach that takes into consideration the above mentioned criteria.

We first pose the problem of finding the optimal graph as a rate-distortion optimization problem defined as

$$\min_{L \in \mathbb{R}^{N \times N}} \mathcal{D}(L) + \gamma(\mathcal{R}_c(L) + \mathcal{R}_G(L)), \tag{3}$$

where $D(L)$ is the distortion between the original image and the reconstructed one. The total coding rate is composed of two representation costs, namely the cost of the transform coefficients $R_c(L)$ and the cost of the graph description $R_G(L)$. Each of these terms possibly depends on $L$ and on the coding scheme. We describe them in more details in the rest of the section.

### B. Distortion approximation

The distortion $\mathcal{D}(L)$ is defined as follows

$$\mathcal{D}(L) = \|x - \tilde{x}\|^2 = \|\hat{x} - \hat{x}_q\|^2,$$

where $x$ and $\tilde{x}$ are respectively the original and the reconstructed image, and $\hat{x}$ and $\hat{x}_q$ are respectively the transform coefficients and the quantized transform coefficients. The equality holds due to the orthonormality of the GFT. Considering a uniform scalar quantizer with the same step size $q$ for all the transform coefficients, if $q$ is small the expected value of the distortion $\mathcal{D}(L)$ can be approximated as follows [10]

$$\mathcal{D} = \frac{q^2 N}{12}.$$

With this approximation, the distortion depends only on the quantization step size and it does not depend on the chosen $L$ [2]. For simplicity, in the rest of the paper we adopt this assumption. Therefore, the optimization problem (3) is reduced to minimizing the rate terms.

### C. Rate approximation of the transform coefficients

We can evaluate the cost of the transform coefficients $R_c(L)$ by using the approximation proposed in [2], [4]

$$\mathcal{R}_c(L) = x^T L x = x^T \left( \sum_{l=0}^{N-1} \lambda_l \psi_l \psi_l^T \right) x$$

$$= \sum_{l=0}^{N-1} \lambda_l (x^T \psi_l)(\psi_l^T x) = \sum_{l=0}^{N-1} \lambda_l \hat{x}_l^2, \tag{4}$$

where $\lambda_l$ and $\psi_l$ are respectively the $l$-th eigenvalue and eigenvector of $L$. Therefore, $\mathcal{R}_c(L)$ is an eigenvalue-weighted sum of squared transform coefficients. It assumes that the
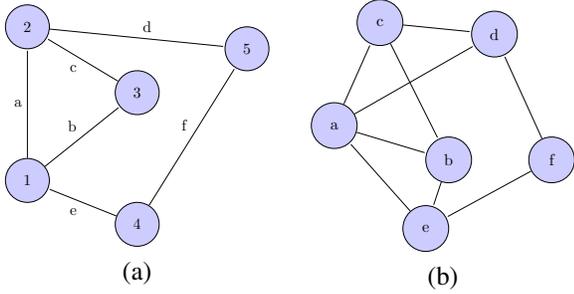
Fig. 1. An example of a graph (a) and its corresponding dual graph (b). The edges in the first graph (labeled with lower case letters) become the nodes of the corresponding dual graph.

coding rate decreases when the smoothness of the signal $x$ over the graph defined by $L$ increases. In addition, (4) relates the measure of the signal smoothness with the sparsity of the transform coefficients. The approximation in (4) does not take into account the coefficients that corresponds to $\lambda_0 = 0$ (i.e., the DC coefficients). Thus, (4) does not capture the variable cost of DC coefficients in cases where the graph contains a variable number of connected components. However, we impose that the graph is connected in our work, which removes the influence of the number of connected components. In this case, the cost of the DC coefficient is independent of $L$, so we can avoid to consider it in the $\mathcal{R}_c$ approximation.

### D. Rate approximation of the graph description

The graph description cost $\mathcal{R}_G(L)$ depends on the chosen method to code the graph. In order to reduce the graph transmission cost, we choose to use a fixed topology $B$ for the graph and to vary only the edge weights. Therefore, the graph can be defined only by a vector $w \in \mathbb{R}^M$, where $w_e$ is the weight of the edge $e$. Then, by using (1) we can define the graph Laplacian $L = B^T \text{diag}(w) B$.

In order to compress the edge weight vector $w$, we propose to treat it as a graph signal that lies on the dual graph $\mathcal{G}_d$. Given a graph $\mathcal{G}$, its dual graph $\mathcal{G}_d$ is an unweighted graph where each node of $\mathcal{G}_d$ represents an edge of $\mathcal{G}$ and two nodes of $\mathcal{G}_d$ are connected if and only if their corresponding edges in $\mathcal{G}$ share a common endpoint. An example of a dual graph is shown in Fig. 1. We choose to use this graph representation for the signal $w$ because consecutive edges usually have similar weights, so the dual graph can provide a smooth representation of $w$. Also for $\mathcal{G}_d$ we can define its graph Laplacian matrix $L_d \in \mathbb{R}^{M \times M}$ and the corresponding eigenvector and eigenvalue matrices $\Psi_d \in \mathbb{R}^{M \times M}$ and $\Lambda_d \in \mathbb{R}^{M \times M}$ such that $L_d = \Psi_d \Lambda_d \Psi_d^T$.

Since $w$ can be represented as a graph signal, we can compute its GFT $\hat{w} \in \mathbb{R}^M$ as

$$\hat{w} = \Psi_d^T w.$$

Therefore, we can use $\hat{w}$ to describe the graph and we evaluate the cost of the graph description by measuring the coding cost of $\hat{w}$. It has been shown that the total bit budget needed to code a vector is proportional to the number of non-zero coefficients

[11], thus we approximate the cost of the graph description by measuring the sparsity of $\hat{w}$ as follows

$$\mathcal{R}_G = \|\hat{w}\|_1 = \|\Psi_d^T w\|_1. \tag{5}$$

### E. Graph learning problem

By using (1), (4) and (5), the graph learning problem (3) becomes equivalent to the following optimization problem

$$\min_{w \in \mathbb{R}^M} x^T B(\text{diag}(w)) B^T x + \alpha \|\Phi^T w\|_1, \tag{6}$$

where $\alpha$ is a constant parameter.

Building on the rate-distortion formulation of (6), we find the optimal graph topology by solving the following optimization problem

$$\min_{w \in \mathbb{R}^M} x^T B(\text{diag}(w)) B^T x + \alpha \|\Phi^T w\|_1 - \beta \mathbf{1}^T \log(w),$$
$$\text{s. t.} \quad w \leq \mathbf{1}, \tag{7}$$

where $\alpha$ and $\beta$ are two positive regularization parameters and $\mathbf{1}$ denotes the constant one vector. The logarithmic term has been added to penalize low weight values and to avoid the trivial solution. In addition, this term guarantees that $w_m > 0$, $\forall m$, so that the graph is always connected. The inequality constraint has been added only to guarantee that all the weights are in the range $(0, 1]$, which is the same range of the most commonly used weighting functions [12].

The problem in (7) can be cast as a convex optimization problem with a unique minimizer. To solve problem (7), we write the first term in the following form

$$x^T B(\text{diag}(w)) B^T x = \text{tr}((B^T x x^T B)\text{diag}(w))$$
$$= \text{vec}(B^T x x^T B)^T \text{vec}(\text{diag}(w))$$
$$= \text{vec}(B^T x x^T B)^T M_{\text{diag}} w,$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix, $\text{vec}(\cdot)$ is the vectorization operator, and $M_{\text{diag}} \in \mathbb{R}^{M^2 \times M}$ is a matrix that converts the vector $w$ in $\text{vec}(\text{diag}(w))$. Then, we can rewrite problem (7) as

$$\min_{w \in \mathbb{R}^M} \text{vec}(B^T x x^T B)^T M_{\text{diag}} w + \alpha \|\Psi_d^T w\|_1 - \beta \mathbf{1}^T \log(w),$$
$$\text{s. t.} \quad w \leq \mathbf{1}. \tag{8}$$

The problem in (8) is a convex problem with respect to the variable $w$ and can be solved efficiently via interior-point methods [13].

### IV. IMAGE COMPRESSION APPLICATION

We now describe how the above graph learning problem can be applied to image compression. As pointed out in the previous sections, we have two different information to transmit to the decoder: the transform coefficients of the image signal $\hat{x}$ and the description of the graph $\hat{w}$. The transform coefficients are quantized using a uniform quantizer with the same step size $q$ for all the coefficients. Then, we code the quantized coefficients until the last non-zero coefficient using an adaptive bitplane arithmetic encoder and we transmit the position of the last significant coefficient.
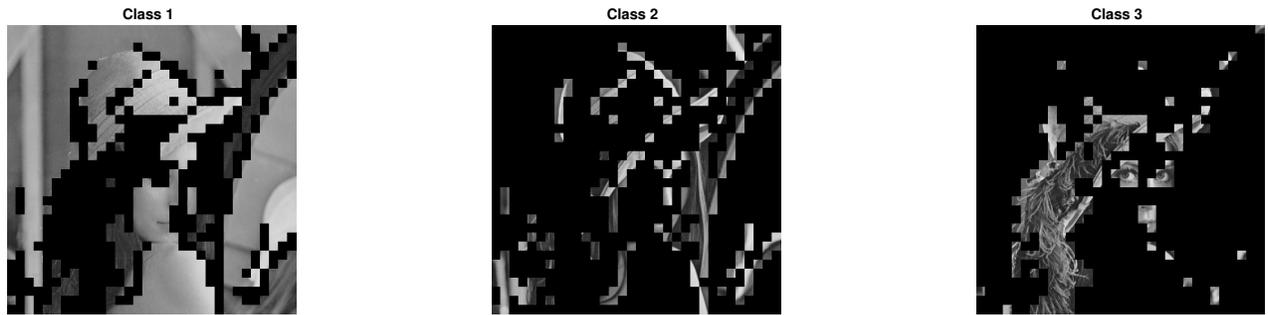
Fig. 2. Block classification of Lena.

To code the graph, we use its GFT coefficients vector $\hat{w}$. In order to reduce the cost of the graph description, we reduce the number of elements in $\hat{w}$ taking into account only the first $\widetilde{M} \ll M$ coefficients, which usually are the most significant, and setting the other $M - \widetilde{M}$ coefficients to zero. The reduced vector $\hat{w}_r \in \mathbb{R}^{\widetilde{M}}$ is then quantized and coded with the same entropy coder used for the image signal.

The principal steps of the proposed image compression method are summarized in Fig. 3. Given an image signal, we first solve the optimization problem in (8) obtaining the optimal solution $w^*$. To transmit $w^*$ to the decoder, we first compute its GFT coefficients $\hat{w}^*$ and the reduced vector $\hat{w}_r^*$, then we quantize and code it using an entropy coder. It is important to underline that, since we perform a quantization of $\hat{w}_r^*$, the reconstructed signal $\tilde{w}^*$ is not equal to the original $w^*$ and its quality depends on the quantization step size used. The graph described by $\tilde{w}^*$ is then used to define the GFT transform for the image signal.

Since it is important to find the best tradeoff between the quality of the graph and its transmission cost, for each block we test different quantization step sizes $\{\Delta_i\}_{1 \leq i \leq Q}$ for a given graph represented by $\hat{w}_r^*$. To choose the best quantization step size, we use the following rate-distortion problem

$$\min_i \mathcal{D}(\Delta_i) + \gamma(\mathcal{R}_c(\Delta_i) + \mathcal{R}_G(\Delta_i)), \qquad (9)$$

where $\mathcal{R}_G(\Delta_i)$ is the rate of $\hat{w}_{r,\Delta_i}^*$, the coefficient vector $\hat{w}_r^*$ quantized with $\Delta_i$, $\mathcal{D}(\Delta_i)$ and $\mathcal{R}_c(\Delta_i)$ are respectively the distortion and the rate of the reconstructed image signal obtained using the graph transform described by $\hat{w}_{r,\Delta_i}^*$. We underline that in (9) we evaluate the actual distortion and rate without using the approximation introduced previously in (3), (4), (5). The coding methods described previously are used to compute the rates $\mathcal{R}_c(\Delta_i)$ and $\mathcal{R}_G(\Delta_i)$.

## V. Experimental results

In this section, we evaluate the performance of the proposed method. We first describe the general experimental setting, then we present the experimental results obtained.

### A. Experimental setup

We test our method on four standard grayscale images (Lena, Boat, Peppers and House) and we split them into non-overlapping $16 \times 16$ pixel blocks. The chosen topology
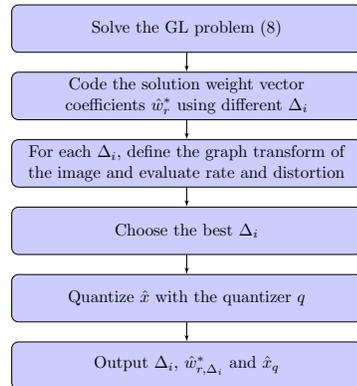


Fig. 3. Block diagram of the proposed method.

of the graph is a 4-connected grid: this is the most used graph topology for graph-based image compression, since its number of edges is not too high, and thus the coding cost is limited. In a 4-connected square grid with $N$ nodes, we have $M = 2\sqrt{N}(\sqrt{N} - 1)$ edges. In all our experiments, we set $\widetilde{M} = 64$ and $Q = 8$. To find the best value for the parameters $\alpha$ and $\beta$ of the graph learning problem in (8), we use the following strategy. The value of the parameter $\alpha$ depends on the characteristics of the block. For this reason, we perform a block classification using the structure tensor analysis, as done in [14]. Let $\mu_1$ and $\mu_2$ be the two eigenvalues of the structure tensor, where $\mu_1 \geq \mu_2 \geq 0$, we can subdivide the image blocks in the following way:

- Class 1: smooth blocks, if $\mu_1 \approx \mu_2 \approx 0$;
- Class 2: blocks with a dominant principal gradient, if $\mu_1 \gg \mu_2 \approx 0$;
- Class 3: blocks with a more complex structure, if $\mu_1$ and $\mu_2$ are both large.

Fig. 2 shows an example of block classification. We set $\alpha = 100$ for blocks that belong to the first class, $\alpha = 500$ for blocks that belong to the second class and $\alpha = 800$ for blocks that belong to the third class. For all the three classes, we set $\beta = 1$ in the graph learning problem.

We compare the performance of the proposed method against the classical DCT transform. To have a fair comparison, we code the transform coefficients $\hat{x}$ of the image signal using the same entropy coder for the graph-based method and for DCT-based encoder. In the first case, in addition to the
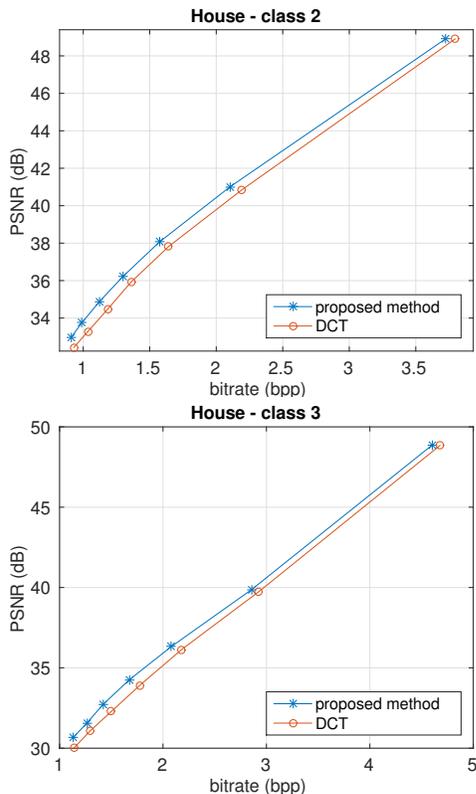
Fig. 4. RD comparison between the proposed method and DCT.

TABLE I
AVERAGE GAIN IN PSNR MEASURED WITH THE BJONTEGAARD METRIC.

| Image | class 1 | class 2 | class 3 |
|---|---|---|---|
| Lena | 0.06 | 0.66 | 0.52 |
| Boat | 0.03 | 0.31 | 0.47 |
| Peppers | 0.10 | 0.77 | 0.89 |
| House | 0.02 | 0.63 | 0.62 |

bitrate of $\hat{x}$, we count the bitrate due to the trasmission of $\hat{w}^*_{r,\Delta_i}$ and 3 additional bits per block to transmit the chosen quantization step size $\Delta_i$ for $\hat{w}_r$. For both methods, we vary the quantization step size $q$ of the transform coefficients to vary the encoding rates.

Finally, in our method for each block we compare the RD-cost of the GFT and the one of the DCT. Then, we eventually code the block with the transform that has the lowest RD-cost and we use 1 additional bit per block to signal if we are using the GFT or the DCT.

*B. Results*

In Fig. 4, we show the performance of the two methods on the image House. More results are given in Table I, where we use the Bjontegaard metric [15] to compute the average gain in PSNR compared to the DCT. In the second and third classes, the proposed method outperforms DCT providing an average PSNR gain of 0.6 dB for blocks in the second class and 0.64 dB for blocks in the third class. It is interesting to point out that there is not a significant difference in performance between the second class and the third one. Instead, in the first class the gain is nearly 0, as DCT in this case is already optimal.

The obtained results show that the proposed method can outperform classical fixed transforms as DCT, even if they could be further improved by optimizing the coding method, in particular the quantization strategy.

## VI. CONCLUSION

In this paper, we have introduced a new graph-based framework for image compression. We have proposed an innovative method for coding the graph by treating the edge weights as a new signal that lies on the dual graph. In order to obtain an effective coding method, we have also formulated a graph learning problem targeted for image compression. The solution of the proposed learning problem is a graph that provides an effective tradeoff between the quality of the transform and the cost of the graph description.

We believe that the proposed method participates to opening a new research direction in graph-based image compression. The obtained results show that the proposed method outperforms the classical DCT. More work on the coding part may lead to further improvements in coding performance.

## ACKNOWLEDGMENT

## REFERENCES

[1] D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
[2] W. Hu, G. Cheung, A. Ortega, and O. C. Au, "Multiresolution graph fourier transform for compression of piecewise smooth images," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 419–433, 2015.
[3] G. Shen, W. S. Kim, S. K. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *Proc. Picture Coding Symposium (PCS)*, 2010, pp. 2808–2811.
[4] W. Kim, S. K. Narang, and A. Ortega, "Graph based transforms for depth video coding," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 813–816.
[5] E. Pavez and A. Ortega, "Generalized laplacian precision matrix estimation for graph signal processing," in *Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2016.
[6] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," *arXiv preprint arXiv:1406.7842*, 2014.
[7] E. Pavez, H. E. Egilmez, Y. Wang, and A. Ortega, "GTT: Graph template transforms with applications to image coding," in *Proc. Picture Coding Symposium (PCS), 2015*. IEEE, 2015, pp. 199–203.
[8] J. Gallier, "Elementary spectral graph theory applications to graph clustering using normalized cuts: a survey," *arXiv preprint arXiv:1311.2492*, 2013.
[9] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *Proc. ICML workshop on statistical relational learning and Its connections to other fields*, vol. 15, 2004, pp. 67–68.
[10] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, 1998.
[11] S. Mallat and F. Falzon, "Analysis of low bit rate image transform coding," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1027–1042, 1998.
[12] L. J. Grady and J. Polimeni, *Discrete calculus: Applied analysis on graphs for computational science*. Springer Science & Business Media, 2010.
[13] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
[14] I. Rotondo, G. Cheung, A. Ortega, and H. E. Egilmez, "Designing sparse graphs via structure tensor for block transform coding of images," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2015, pp. 571–574.
[15] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, 2-4 April 2001*, 2001.