SPARSE INVERSE BILATERAL FILTERS FOR IMAGE PROCESSING

Akshay Gadde¹, Mengying Xu² and Antonio Ortega³

^{1,3} University of Southern California, Los Angeles
 ² Tsinghua University, Beijing, China

ABSTRACT

The bilateral filter (BF) is a prominent tool for adaptive, structurepreserving image filtering. It can be interpreted as a graph-based filter, where the nodes of the graph correspond to image pixels and link weights correspond to filter coefficients. Graphs associated to BFs of typical sizes used in practice are very dense. In this paper, we propose an efficient method for constructing a sparse graph for adaptive filtering of an image. The Laplacian matrix of the proposed sparse graph approximates the inverse of a dense BF kernel matrix. This is analogous to the idea of finding a sparse inverse covariance of a Gaussian Markov random field (GMRF) with a dense covariance matrix. The eigenvectors of the proposed graph Laplacian are approximately equal to the eigenvectors of the BF graph and allow for low frequency representation of the image similar to the BF eigenvectors. Filters in the form of polynomials of this sparse Laplacian offer a more flexible and less computationally complex alternative to a dense BF, with similar performance.

Index Terms— Bilateral filter, Graph signal processing, Sparse graph construction, Gaussian Markov random field

1. INTRODUCTION

Natural images have complex structure due to the presence of textures and of edges with different orientations. Filtering such images with fixed non-adaptive filters tends to blur these details as it involves averaging of pixels across discontinuities. In order to account for the complexities in natural images, modern filtering and representation methods use image adaptive processing. Examples of such filters include the bilateral filter (BF) [1], anisotropic diffusion, nonlocal means [2] or kernel regression [3]. Adaptively designed bases also allow more compact representation of images. Graph signal processing (GSP) [4] offers a principled framework for adaptive image processing. In the GSP formulation, an image is represented as a graph, where the nodes correspond to pixels and links connecting the nodes capture the similarities between pixels. The Laplacian matrix associated with the graph can then be used to design structure preserving filters [3, 5]. The so-called graph Fourier transform (GFT), defined using the eigenvectors of the Laplacian, allows for compact representation in which most of the signal energy is captured in the low-frequency GFT coefficients [6, 7].

In this paper, we focus on the bilateral filter, which is a prominent tool for adaptive image processing, widely used in different applications such as denoising, edge preserving multiscale decomposition, segmentation etc. [8, 9]. The BF can be interpreted as a filter on a graph in which the similarity weights associated with links are given by the filter coefficients [5]. The GFT defined using this BF graph also allows for compact low frequency representation of the image. The BF can be expressed as a low pass filter in this GFT domain. In the graph associated with the $k \times k$ BF, each node is connected to k^2 neighbors. For large values of k (k = 5, 7, 9 are commonly used), such a graph is very dense. Computational complexity of a single application of BF is roughly $O(mnk^2)$, where $m \times n$ is the image size. Using such dense BF graph to apply additional GSP tools (such as graph wavelets [6] or graph based regularization [5, 10]) for adaptive image processing can be even more computationally complex.

Motivated by this, we consider the following question: *Is it possible to give a dense BF graph a sparse graph representation that has similar eigenstructure and offers similar filtering performance, but with lower complexity?* We propose to construct such a sparse graph by approximating the inverse of the BF kernel matrix in the form of a generalized Laplacian (GL, see Section 2). Eigenvectors of this GL approximate the eigenvectors to the BF graph (because a matrix and its inverse have the same eigenvectors). Low pass polynomial filters on this graph offer comparable performance to a dense BF with less numerical complexity.

Several methods [11, 12, 13] have been proposed to estimate a sparse inverse of a dense positive definite matrix in the context of precision matrix estimation of a Gaussian Markov random field (GMRF). We use the method proposed in [14] since it restricts the inverse to be in the form of a GL. Because of the positive definiteness of the BF kernel matrix [15], it is a valid input to this GL estimation algorithm. If the matrix to be inverted is a covariance matrix of a GMRF, then the GL estimates the corresponding precision (or inverse covariance) matrix. The precision matrix of a GMRF captures the conditional independence relations between the variables [16] and is expected to be sparse. The BF kernel function can be interpreted as a covariance between two pixels that decays rapidly as the geometric or photometric distance between the pixels increases. Under this analogy, each pixel is expected to conditionally independent of other pixels, given the pixels which are most similar to it [17]. Therefore, the inverse of the BF kernel matrix, which is analogous to a precision matrix, is expected to be sparse.

We propose a simple heuristic algorithm to efficiently approximate the estimated GL based on the observed conditional independence structure. The proposed heuristic voids the need for actually estimating a GL using the method in [14]. The approximate GL given by our heuristic is very sparse, such that each node (i.e., pixel) is connected to roughly only 4 other nodes. We show empirically that the approximate GL offers low frequency representation of the image in its GFT basis, which is more compact than the one obtained using a dense 7×7 BF graph. This is further demonstrated by the superior denoising performance of the Wiener filter defined using the GFT of the approximate GL. We design polynomial filters using the approximate GL that offer comparable performance to the 7×7 BF at lower computational cost.

This work was supported in part by NSF under grants CCF-1410009 and CCF-1527874.

2. MATHEMATICAL PRELIMINARIES

2.1. A Brief Review of GSP

Consider an undirected and weighted graph G with nodes \mathcal{V} indexed by $\{1, 2, \ldots, N\}$ and the edge set $\mathcal{E} = \{(i, j, w_{ij})\}$, where (i, j, w_{ij}) denotes an edge with weight w_{ij} connecting nodes i and j. The connectivity information is encoded by the adjacency matrix \mathbf{W} of size $N \times N$. The degree matrix $\mathbf{D} = \text{diag}\{d_1, \ldots, d_N\}$, where $d_i = \sum_j w_{ij}$ is the degree of node i. The Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The symmetric normalized forms of the adjacency and the Laplacian are given by $\mathcal{W} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ and $\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. A graph signal $f : \mathcal{V} \to \mathbb{R}$ is a mapping that takes a real value on each node of the graph and can be represented as $\mathbf{f} = [\mathbf{f}_1, \ldots, \mathbf{f}_N]^\top \in \mathbb{R}^N$.

L (and \mathcal{L}) has real eigenvalues $0 \leq \lambda_1 \leq \ldots \leq \lambda_N$ and a corresponding orthogonal set of eigenvectors $\mathbf{U} = [\mathbf{u}^1, \ldots, \mathbf{u}^N]$. The Graph Fourier Transform (GFT) of a signal **f** is defined as $\mathbf{\tilde{f}}_i = \langle \mathbf{f}, \mathbf{u}^i \rangle$ (or in an equivalent matrix form $\mathbf{\tilde{f}} = \mathbf{U}^{\top} \mathbf{f}$), where $\mathbf{\tilde{f}}_i$ is the GFT coefficient corresponding to frequency λ_i [4]. The GFT can also be defined using \mathcal{L} .

Spectral domain filtering of graph signals can be defined using the GFT [18]. These spectral filters are of the form $\mathbf{U}h(\mathbf{\Lambda})\mathbf{U}^{\mathsf{T}}$, where $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \ldots, \lambda_N\}$. They modulate the GFT coefficients of the input signal by the spectral response $h(\lambda)$ to produce the output. If $h(\lambda)$ is a polynomial $\sum_{i=0}^{p} a_i \lambda^i$, then the filter can be written as $\sum_{i=0}^{p} a_i \mathbf{L}^i$. Implementation of such polynomial filters boils down to computing p matrix–vector multiplications of the form \mathbf{Lx} . A non-polynomial spectral filter can be approximated by a polynomial filter of sufficient degree for efficient implementation.

2.2. Graph Based Interpretation of the BF

A graph based interpretation of the BF can be obtained with nodes corresponding to pixels and edge weights given by [5]:

$$w_{ij} = \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\sigma_d^2}\right) \exp\left(-\frac{(\mathbf{f}_i - \mathbf{f}_j)^2}{2\sigma_r^2}\right), \quad (1)$$

where \mathbf{p}_i denotes the position of pixel *i* and \mathbf{f}_i is the pixel intensity. In a $k \times k$ BF graph, nodes *i* and *j* are connected iff *j* is in the $k \times k$ neighborhood of pixel *i*. An example 5×5 BF graph corresponding to block 2 in Figure 3 is shown in Figure 1(a).

With **f** as the graph signal, the BF can be written as $\mathbf{y} = \mathbf{D}^{-1}\mathbf{W}\mathbf{f}$. Defining $\hat{\mathbf{f}} = \mathbf{D}^{1/2}\mathbf{f}$, we can rewrite the BF as $\hat{\mathbf{y}} = (\mathbf{I} - \mathcal{L})\hat{\mathbf{f}}$. Thus, the BF can be interpreted as a low pass spectral filter in the GFT domain defined using \mathcal{L} with $h(\lambda) = 1 - \lambda$ [5]. We propose to extend this idea by defining polynomial filters on a sparse graph whose GFT approximates that of the original dense BF graph. These filters will offer comparable or better performance to the dense BF, but with lower complexity.

2.3. Laplacian Based Smoothness and GMRF

The notion of frequency defined for graph signals using the Laplacian can be given a probabilistic interpretation by assuming that the signals follow a GMRF model [19, 20]:

$$p(\mathbf{x}) \propto \exp\left(-\sum_{i,j} q_{ij}(\mathbf{x}_i - \mathbf{x}_j)^2 - \sum_i q_{ii}\mathbf{x}_i^2\right).$$
 (2)

The term in the exponent can be rewritten as $-\mathbf{x}^{\top}\mathbf{Q}\mathbf{x}$, where \mathbf{Q} is the inverse covariance (or precision) matrix. If \mathbf{Q} is a symmetric

positive semi-definite matrix of the form $\alpha \mathbf{I} - \mathbf{N}$, where \mathbf{I} is the identity matrix and $\mathbf{N}_{ij} \geq 0 \forall i, j$, then it is called a generalized Laplacian (GL). Note that if $q_{ii} = \sum_j -q_{ij}$, then \mathbf{Q} is a Laplacian, while in general \mathbf{Q} can be interpreted as a Laplacian with self loops. If $\mathbf{x}^{\top} \mathbf{Q} \mathbf{x}$ is small, then \mathbf{x} will have high likelihood with respect to the GMRF. Therefore, if the GMRF is represented as a graph with Laplacian \mathbf{Q} , then a signal with high likelihood will be smooth on that graph.

For $\mathbf{x} \sim \text{GMRF}$, the conditional correlation between \mathbf{x}_i and \mathbf{x}_j given the rest of the variables $\operatorname{corr}(\mathbf{x}_i, \mathbf{x}_j | \mathbf{x}_{k \neq i,j}) = -q_{ij}/\sqrt{q_{ii}q_{jj}}$. $\mathbf{x}_i, \mathbf{x}_j$ are conditionally independent iff $q_{ij} = 0$. Therefore, \mathbf{Q} is expected to be sparse [16].

An algorithm to estimate the inverse of a positive definite matrix \mathbf{K} in the form of a GL \mathbf{Q} has been proposed in [14]. It solves the following problem:

$$\min_{\mathbf{Q} \succeq \mathbf{0}; \ q_{ij} \le 0, i \neq j} -\log \det(\mathbf{Q}) + \operatorname{tr}(\mathbf{KQ})$$
(3)

If \mathbf{K} is a sample covariance matrix, then the above problem can be thought of as a maximum likelihood estimation problem of a GMRF under GL constraints.

3. SPARSE GRAPH CONSTRUCTION FOR IMAGES

3.1. Inverse of the BF Kernel Matrix as a Sparse GL

In order to find a sparse graph for adaptive filtering of an image, we propose to solve (3) with **K** given by the BF kernel matrix, i.e., $\mathbf{K}_{ij} = w_{ij}$ as defined in (1). Note that the BF kernel matrix is positive definite [15] and hence, is a valid input for (3).

The idea of representing a dense BF kernel matrix by a sparse graph with Laplacian $\mathbf{Q} \approx \mathbf{K}^{-1}$ is analogous to the idea of parametrizing a GMRF, which has a dense covariance matrix, by a sparse precision matrix. If we extend the analogy further and interpret the BF kernel \mathbf{K}_{ij} as covariance between \mathbf{f}_i and \mathbf{f}_j , then the zero entries of the estimated GL \mathbf{Q} can be viewed as conditional independence relations between pixels. Since \mathbf{K}_{ij} decays very rapidly as $\|\mathbf{p}_i - \mathbf{p}_j\|$ or $|\mathbf{f}_i - \mathbf{f}_j|$ increases, it is reasonable to expect \mathbf{f}_i to be conditionally independent of other pixels, given the pixels which are most similar to it [17]. Therefore, we expect to \mathbf{K}^{-1} to be well approximated by a sparse \mathbf{Q} . In our experiments, we observe that \mathbf{Q} obtained by solving (3) is very sparse even with the absence of an explicit sparsity constraint.

Since $\mathbf{Q} \approx \mathbf{K}^{-1}$, eigenvectors of \mathbf{Q} will be approximately equal to the GFT basis of the BF graph (i.e., the eigenvectors of \mathcal{L} with $\mathbf{W} = \mathbf{K}$). Moreover, an increasing order of the graph frequencies of the BF graph (i.e., the eigenvalues of \mathcal{L}) will correspond to a similar ordering of the eigenvalues of \mathbf{Q} . It has been observed that an image has a low frequency representation in the GFT of the BF graph [3, 5]. Therefore, the image will also be low pass in the GFT defined using \mathbf{Q} . Thus, the estimated sparse GL \mathbf{Q} can be used for designing computationally efficient low pass polynomial graph filters (as explained in Section 2.1) with similar performance to the dense BF.

3.2. Efficient Approximation of the Estimated GL

Solving (3) for estimating \mathbf{Q} is computationally not feasible for large images. In fact, our motivation was to reduce computational complexity of filtering and representation by avoiding the computation of a dense BF graph. Then, the goal is to quickly find a sparse graph from an image that has similar eigenstructure as a dense BF graph and enables efficient filtering.



Fig. 1: Different types of graphs corresponding to block 2 in Figure 3.



Fig. 2: Plot of $-q_{ij}/\max(|\mathbf{Q}|)$ vs. \mathbf{K}_{ij} for one pixel in block 2

To achieve this goal, we propose a simple and efficient heuristic for approximating the GL estimated by (3). The proposed heuristic is based on the following observed characteristics of the estimated GLs (see Figure 1 for an example): 1. Each node is connected to roughly four other nodes. 2. At each node *i*, only the connections with the highest BF similarity \mathbf{K}_{ij} are preserved in \mathbf{Q} . An example is shown in Figure 2. 3. Most of the connections are between nodes which are within 2-hop distance of each other. The last observation is consistent with the second, since the BF similarity between nodes which are far from each other will be small.

These observations have a nice interpretation similar to the Markov property of GMRFs [16]: the signal value \mathbf{f}_i at node *i* is conditionally independent of the signal at other nodes given the 4 nodes which are most similar to *i*. The observations suggest the heuristic shown in Algorithm 1 for fast approximation of \mathbf{Q} .

Algorithm 1 Heuristic for fast approximation of Q

1: Input: Image $\mathbf{f} \in \mathbb{R}^N$ 2: Initialize: $\mathbf{W} = \mathbf{0} \ (\in \mathbb{R}^{N \times N})$ 3: for all pixels *i* do 4: Compute \mathbf{K}_{ij} with (1) $\forall j \in 5 \times 5$ neighborhood of *i* 5: Keep the largest 4 entries in \mathbf{W} and set the rest to 0 6: end for 7: $\mathbf{W} \leftarrow (\mathbf{W} + \mathbf{W}^{\top})/2$ 8: $d_i := \sum_j \mathbf{W}_{ij} \ \forall i$ and $\mathbf{D} := \text{diag}\{d_i\}$ 9: Output: $\mathbf{Q}_{\text{approx}} := \mathbf{D} - \mathbf{W}$

Although only the entries corresponding 4 largest values \mathbf{K}_{ij} are preserved in \mathbf{Q} for each node *i*, the mapping $\mathbf{K}_{ij} \mapsto \mathbf{Q}_{ij}$ can



Fig. 3: Different image blocks taken from "Lena" image

be very non-linear as can be seen in Figure 2. We plan to investigate this mapping in future for a better approximation of \mathbf{Q} .

4. REPRESENTATION AND FILTERING WITH PROPOSED GRAPH

We now demonstrate experimentally the efficacy of the proposed sparse graph construction in image representation and filtering. We consider four blocks taken from the "Lena" image, which are shown in Figure 3. The blocks are selected in order to capture different kinds of patterns such as strong edges, textures and smooth regions commonly observed in natural images. To compare the quality of representation offered by different graphs, we compute the fraction of energy captured in the first m GFT coefficients, $\sum_{i=1}^{m} \tilde{\mathbf{f}}_{i}^{2} / \|\mathbf{f}\|^{2}$, obtained using the respective graph Laplacians. Figure 4 shows that the GL and its approximation obtained using Algorithm 1 (AGL) offer better energy compaction in the low frequency GFT coefficients than the BF graph, especially when the image has more discontinuities, as is the case in blocks 1 and 4. Moreover, the proposed graph construction methods produce much sparser graphs than the conventional BF graph. Average number of non-zero entries in the 7×7 BF graph, the GL and AGL are 8649, 1021 and 1168, respectively.

In order to evaluate the filtering performance of different graphs, we consider denoising as an example application. We observe a noisy version of the original image, corrupted by i.i.d. Gaussian noise. The observed image is filtered using the following different filters: 1. 7×7 BF; 2. Wiener filter in the GFT basis of the BF graph; 3. Wiener filter in the GFT basis obtained using the GL; 4. Wiener filter in the GFT basis obtained using AGL; 5. A polynomial filter defined using AGL.

Wiener filter is the optimal filter for minimizing the MSE. Its spectral response [3] is given by $h^{\rm w}(\lambda_i) = \tilde{\mathbf{f}}_i^2/(\tilde{\mathbf{f}}_i^2 + \sigma_n^2)$, where σ_n^2 is the noise variance. Note that it is a nonlinear filter which requires



Fig. 4: Fraction of energy in first m GFT coefficients. The plot corresponding to block 3 is omitted, since GFTs of all graphs offer similar energy compaction.



Fig. 5: Spectral response of a polynomial filter using approximate GL

explicit computation of the GFT. The polynomial filter in Q_{approx} approximates a filter with spectral response

$$h(\lambda) = \frac{1 + \exp(-rc)}{1 + \exp\left(r(\lambda/\lambda_{\max} - c)\right)}.$$
(4)

It is a linear low pass filter with cutoff $c\lambda_{\text{max}}$. r controls the rate of decay (see Figure 5). The response does not depend on the signal. We approximate $h(\lambda)$ with Chebyshev polynomials [18] of degree 8. Since most of the signal energy is captured by the low frequency GFT coefficients and noise is spread in all frequencies, a low pass filter will attenuate most of the noise while preserving the signal. Choice of the cutoff depends on the noise level.

The PSNR values of the results for different images at various noise levels are shown in Table 1. The Wiener filters on GL and AGL show slightly better results than the Wiener filter on BF graph whenever the representation of the clean signal is much more compact in the GFT basis of GL than the GFT of BF graph. The results are comparable in other cases. Polynomial filters in the AGL also offer slightly better or comparable results than the 7×7 BF.

As explained earlier, complexity of a $k \times k$ BF is $O(mnk^2)$. On the other hand, consider a polynomial filter of degree p operating on a graph where each node (i.e., pixel) is connected to at most 4 other nodes. Such a filter can be implemented as a sequence of pdegree-1 filters, each with complexity 4mn, followed by a sum of p filtered images. Thus, its total complexity is roughly O(5mnp). Thus, a $(k \times k)$ BF will have roughly the same complexity as a degree $k^2/5$ polynomial filter on a 4-connected graph. In the approximate GL, each node is connected to roughly 4 other nodes. Therefore, a degree 8 polynomial filter in the AGL will have less complexity than the 7 × 7 BF (since 8 < $7^2/5$).

В	Ν	7×7	$7 \times 7 \text{ W}$	GLW	AGLW	AGLP
1	15	21.99	26.09	26.92	27.11	22.68
	20	24.73	29.65	31.04	31.00	24.76
	25	27.94	33.98	34.55	34.45	28.02
	30	31.45	38.81	38.90	38.68	31.01
2	15	23.34	28.07	28.35	28.92	24.70
	20	27.03	32.49	31.66	31.79	27.45
	25	30.71	36.05	35.33	35.91	30.79
	30	34.00	40.47	40.07	39.77	33.83
3	15	25.21	39.88	37.97	39.58	29.93
	20	30.04	42.26	41.14	41.73	34.77
	25	34.65	43.85	43.06	43.19	36.57
	30	38.76	45.51	45.16	45.52	42.09
4	15	22.60	25.82	26.48	26.59	22.96
	20	24.85	29.57	29.72	30.11	23.98
	25	27.26	33.64	34.00	34.15	26.93
	30	30.61	38.57	38.17	38.68	30.43

Table 1: Denoising results. Column 'B' shows the block numbers as specified in Figure 3. All other columns show PSNRs in dB with different filters. (N) noisy input; $(7 \times 7) 7 \times 7$ BF; $(7 \times 7 W)$ Wiener filter on the BF graph; (GLW) Wiener filter on the estimated GL; (AGLW) Wiener filter on the approximate GL; (AGLP) Polynomial filter on the approximate GL.

5. CONCLUSION

We proposed an efficient method for estimating a sparse graph from an image for its adaptive filtering and representation. The proposed method is analogous to the framework of sparse inverse covariance estimation for a GMRF model, where the BF kernel matrix plays the role of a covariance matrix. The Laplacian polynomial filters on this sparse graph offer a less computationally complex alternative to a dense BF with comparable performance. Its GFT basis allows for a more compact low pass representation of the image than the GFT defined using a dense BF graph.

In future, we would like to consider the problem of designing good polynomial filters on the sparse image graphs, based on the graph spectrum and the application at hand. Comparing the performance and complexity of such filters with previously proposed fast approximations of the BF [21, 22] would also be useful. For theoretical understanding of the proposed method, it would be interesting to look at a more general problem of graphical model estimation in reproducing kernel Hilbert spaces.

6. REFERENCES

- C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *International Conference on Computer Vision*, Jan 1998, pp. 839–846.
- [2] A. Buades, B. Coll, and J.M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [3] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106–128, Jan. 2013.
- [4] D. I. Shuman, S. K. 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.
- [5] A. Gadde, S. K. Narang, and A. Ortega, "Bilateral filter: Graph spectral interpretation and extensions," in *IEEE International Conference on Image Processing*, 2013, pp. 1222–1226.
- [6] S. K. Narang, Y.-H. Chao, and A. Ortega, "Graph-wavelet filterbanks for edge-aware image processing," in *IEEE Statistical Signal Processing Workshop*, 2012, pp. 141–144.
- [7] S. K Narang and A. Ortega, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4673– 4685, 2013.
- [8] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "Bilateral filtering: Theory and applications," *Foundations and Trends in Computer Graphics and Vision*, vol. 4, no. 1, pp. 1–73, 2009.
- [9] M. Zhang and B. K. Gunturk, "Multiresolution bilateral filtering for image denoising," *Image Processing, IEEE Transactions on*, vol. 17, no. 12, pp. 2324 –2333, Dec 2008.
- [10] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1047–1060, 2008.
- [11] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
- [12] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the lasso," *The Annals of Statistics*, vol. 34, no. 3, pp. 1436–1462, 06 2006.
- [13] M. J. Wainwright, J. D Lafferty, and P. K. Ravikumar, "Highdimensional graphical model selection using *l*₁-regularized logistic regression," in *Advances in neural information processing systems*, 2006, pp. 1465–1472.
- [14] E. Pavez and A. Ortega, "Generalized laplacian precision matrix estimation for graph signal processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 6350–6354.
- [15] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, pp. 1171–1220, 2008.
- [16] H. Rue and L. Held, Gaussian Markov random fields: theory and applications, CRC Press, 2005.

- [17] F. Lindgren, H. Rue, and J. Lindström, "An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 4, pp. 423–498, 2011.
- [18] D. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129 – 150, 2011.
- [19] A. Gadde and A. Ortega, "A probabilistic interpretation of sampling theory of graph signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 3257–3261.
- [20] C. Zhang, D. Florencio, and P. A. Chou, "Graph signal processing: A probabilistic framework," Tech. Rep., April 2015.
- [21] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 24–52, 2009.
- [22] K. N. Chaudhury, D. Sage, and M. Unser, "Fast O(1) bilateral filtering using trigonometric range kernels," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3376–3382, Dec. 2011.