Spectral Graph Theory and High Performance Computing
Literature Survey

David Wobrock `david.wobrock@gmail.com`

October 26, 2017

1 Related work study on Image Processing using Graph Laplacian Operator

In this section we will explore three topics concerning this project. First of all, we will dive into spectral graph theory. Then, we will deal with 2D image processing, focusing three applications: denoising, sharpening and image segmentation. For each of them, we show a classical algorithm, a new spectral approach and then a quick comparison of them. Finally, we present what has been done in 3D image processing so far using spectral graph theory, essentially using the graph laplacian operator.

1.1 Spectral graph theory

Spectral graph theory has a long history starting with matrix theory and linear algebra that were used to analyse adjancency matrices of graphs. It consists in studying the properties of graphs in relation to the eigenvalues and eigenvectors of the adjancency or Laplacian matrix of the graph. The eigenvalues of such a matrix are called the spectrum of the graph. The second-smallest eigenvalue has been called "algebraic connectivity of a graph" by Fiedler [1], and is therefore also known as *Fiedler value*.

This field is very broad and the eigen-decomposition of graphs is used in a lot of areas. First, it was applied to chemistry because eigenvalues can be associated with the stability of molecules. Spectral graph theory has a many other applications such as graph coloring, graph isomorphism testing, random walks and graph partitioning among others.

One of the most complete works about spectral graph theory is [2] by Fan Chung. This monograph exposes many properties of graphs, the power of the spectrum and how spectral graph theory links the discrete world with the continuous one.

Laplacian matrix Since the adjancency matrix of a graph only holds basic information about it, we usually augment it to the Laplacian matrix. Multiple definitions of the Laplacian matrix are given in [2] and each hold different properties. The most common ones are the normalised Laplacian and the Random walk Laplacian.

Cheeger's inequality One of the most fundamental theorems of spectral graph theory concerns the Cheeger's inequality and Cheeger constant. It approximates the sparsest cut of a graph with the second eigenvalue of its Laplacian.

The Cheeger constant measures the degree of "bottleneck" of a graph, useful for constructing well-connected graphs. Considering a graph G of n vertices, the Cheeger constant is defined as

$$h(G) = min_{0 < |S| \leq \frac{n}{2}} \frac{|\partial S|}{|S|},$$

where S is a subset of the vertices of G and ∂S is the *edge boundary* of S to have all edges with exactly one endpoint in S, or formally

$$\partial S = u, v \in E(G) : u \in S and v \notin S,$$

with $E(G)$ the edges of graph G.

Cheeger's inequality defines a bound and relationship on the smallest positive eigenvalue of the Laplacian matrix $\mathcal{L}$ such as

$$\lambda_1(\mathcal{L}) \geq \frac{h^2(\mathcal{L})}{4}.$$

When the graph G is d-regular, thanks to [3], we also have an inequality between $h(G)$ and the second-smallest eigenvalue λ_2 such as

$$\frac{1}{2}(d - \lambda_2) \leq h(G) \leq \sqrt{2d(d - \lambda_2)}.$$

1.2   2D image processing

1.2.1   Denoising

Background Even with high quality cameras, denoising and improving a taken picture remains important. The two main issues that have to be addressed by denoising are blur and noise. The effect of blur is internal to cameras since the number of samples of the continuous signal is limited and it has to hold the Shannon-Nyquist theorem [4]. Noise comes from the light aquisition system that fluctuates in relation to the amount of incoming photons.

To model these problems, we can formulate the deficient image as,

$$y = z + e,$$

where e is the noise vector of variance σ^2, z the clean signal vector and y the noisy picture.

What we want is a high-performance denoiser, capable of scaling up in relation to increasing the image size and keeping reasonable performances. The output image should come as close as possible to the clean image. As an important matter, it is now accepted that images contain a lot of redundancy. This means that, in a natural image, every small enough window has many similar windows in the same image.

Traditional, patch-based methods   The image denoising algorithms review proposed by [4] suggests that the NL-means algorithm, compared to other reviewed methods, comes closest to the original image when applied to a noisy image. This algorithm takes advantage of the redundancy of natural images and for a given pixel i, predicts its value by using the pixels in its neighbourhood.

In [5], the authors propose the BM3D algorithm, a denoising strategy based on grouping similar 2D fragments of the image into 3D data arrays. Then, collaborative filtering is performed on these groups and return 2D estimates of all grouped blocks. This algorithm exposed state-of-the-art performance in terms of denoising at that time. The results are still one of the best for a reasonable computational cost.

Global filter   In the last couple of years, global image denoising filters came up, based on spectral decompositions [6]. This approach considers the image as a complete graph, where the filter value of each pixel is approximated by all pixels in the image. We define the approximated clean image $\hat{z}$ by,

$$\hat{z} = Wy,$$

where W is our data-dependent global filter, a $n \times n$ matrix, n the number of pixels in the picture. W is computed from the graph affinity matrix[1] K, also of size $n \times n$, such as

$$K = \mathcal{K}_{ij},$$

where $\mathcal{K}$ is a similarity function between two pixels i and j. As the size of K can grow very large, we sample the image and compute an approximated $\hat{K}$ on this subset using the Nyström extension. K can in fact be approximated through its eigenvalues and eigenvectors. Knowing that the eigenvalues of K decay very fast [7], the first eigenvectors of the subset of pixels are enough to compute the approximated $\hat{K}$.

Generally, as proposed in [6] and [8], to improve the denoising performance of global filters, pre-filtering techniques are used. It is proposed to first apply a NL-means algorithm to the image to reduce the noise, but to still compute the global filter on the noisy input and apply the filter to the pre-filtered image.

---

[1]Also called kernel matrix or similarity matrix

Comparison between patch-based and global methods   As [8] suggests, global filter methods have the possibility to converge to a perfect reconstruction of the clean image, which seems to be impossible for techniques like BM3D. Global filtering also seems promising for creating more practical image processing algorithms.

The thesis [9] proposes a normalised iterative denoising algorithm which is patch-based. The work reports that this technique has slightly better results than the global filter but essentially has a better runtime performance.

1.2.2   Sharpening

Background   Sharpening consists basically in a high-pass filter which will magnify high frequency details [9]. The two main issues again with this approach are that noise often displays high frequency attributes, which will be amplified by the filter. Secondly, the phenomenon of overshooting and undershooting, which appears when sharpening already sharp parts (edges for example), will exhibit unpleasant artifacts.

Modeling the problem is the same as for denoising, $\hat{z} = Fy$, F being our filter.

Classical Difference of Gaussian (DoG) operator   This technique consists of computing the difference between two gaussian kernels, which will produce a range of different kernels with various frequencies. Indeed, DoG involves subtracting a blurred version of the input image to another, less blurred version of this original image [10]. A blurred image can be obtained by convolving the input image with a gaussian kernel. This gaussian blurring actually removes high-frequency spatial information. So the idea is that by substracting this blurred image from a less blurred one, we will keep the high-frequency information, like a high-pass filter[2], which results in image sharpening.

Structure-aware sharpening   We know from its definition [11] that the smoothing filter W is a symmetric and doubly stochastic matrix[3]. So the largest eigenvalue $\lambda_1 = 1$ shows that it has low pass filter characteristics. With the definition of the Laplacian, in relation to the matrix W such as $\mathcal{L} = I - W$, we can consider that it is a data-adaptive high-pass filter [9]. So we can define a data-adaptive unsharp mask filter as

$$F_1 = I + \beta(I - W),$$

with $\beta > 0$. This is basically a weighted high-pass filtered version of the input image [7].

But with this approach, the noise amplifying and overshoot problems remain. The proposed solution in [11] applies first a smoothing filter to the image, then the unsharp mask filter and finally the same smoothing filter again.

---

[2]More precisely, like a band-pass filter here

[3]Depending on the used Laplacian definition. For example the one obtained by "Sinkhorn" iterations [12]

The smoothing and sharping filters use possibly two different kernels matrices as basis. The first smoothing filter aims to reduce the noise, while we still avoid over-smoothing. The second and final smoothing controls the effect of the amplified noise and the overshoot artifacts. This gives us the filter

$$F = W_1(I + \beta(I - W_2))W_1,$$

where W_1 and W_2 are constructed from the affinity matrices K_1 and K_2, which can differ in relation to the parameters of their respective kernel function.

Comparison   As [11] suggests, the structure-aware sharpening is data-adaptive and more noise robust than a classical DoG filter. Even though, both approaches use a similar concept. The proposed filter is actually a data-derived DoG-based filter. We can rewrite the proposed filter as

$$F = (1 + \beta)W_1^2 - \beta W_1 W_2 W_1,$$

where we can actually see the difference between two versions of the input image.

1.2.3   Image Segmentation

Background   To recognise patterns, find groups and cut images is a similar problem to graph partitioning. As is it known, most formulations of partitioning a graph result in a NP-hard problem. So it will be our goal to approximate quickly good partitions instead of looking for an optimal solution.

A graph $G = (V, E)$ is partitioned into two disjoint sets A and B such as $A \cup B = V$ and $A \cap B = \emptyset$ by removing edges between A and B. One can compute the value of such a *cut* with the degree of dissimilarity between A and B, which is calculated from the total weight of the removed edges

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v).$$

Minimum cut   The most common way of partitioning a graph is the *minimum cut*. This problem consists in looking for the optimal bipartition of a graph, which is one that minimises the *cut* value. The *minimum cut* problem is well-studied and has efficient algorithms for solving it, even though it has an exponential amount of partitions that can be explored.

Some work on image segmentation using the *minimum cut* has been done [13] [14] [15]. But the results can only be qualified as proposals for recognised regions, such does [14] and would need further optimisation to be the final groups. For instance, [13] introduced a cut criterion using *minimum cut* on a graph but it was observed that it favors finding too small components and [15] tries to optimise the latter.

However, this problem still remains and *minimum cut* can produce bad groups on certain cases. Additionnaly, these cases often appear in natural images, where it is more important to group a large area of the background before small details.
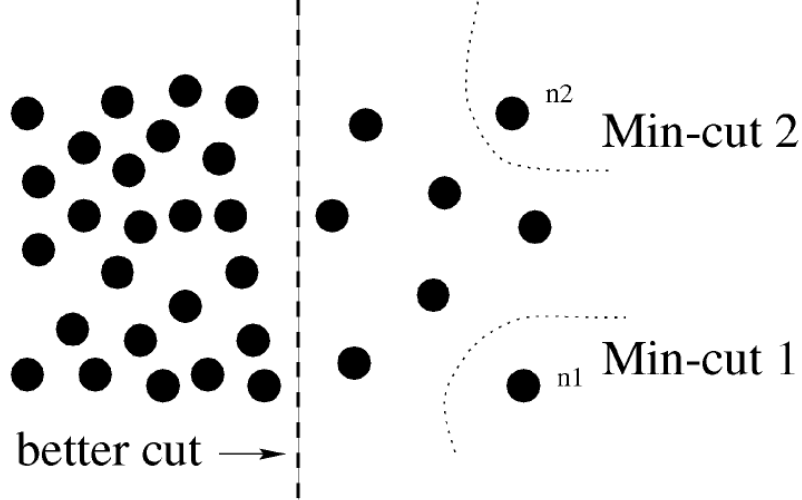
Figure 1: Bad partitioning from *minimum cut* in a particular case [16]

Normalised cut To improve this state, [16] proposed, instead of looking at the total weigh of the edges connecting two groups, to look at a fraction of the total edge connections to all nodes. This is the *normalised cut*, or *Ncut*. The disassociation measure can be formulated as

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)},$$

where $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$, the total connection from the nodes in A to all nodes in the graph.

The goal is still to minimise the *Ncut* value, but when *cut* is small for isolated pixels, the fraction will be greater because the similarity *assoc* between this one pixel and all pixels will be small. This fraction's denominator will be bigger if the similarity between the selected pixels and all pixels is more important, which makes the fraction smaller.

As it follows, we can define the normalised association measure within groups, such as

$$Nassoc(A, B) = \frac{assoc(A, A)}{assoc(A, V)} + \frac{assoc(B, B)}{assoc(B, V)},$$

where $assoc(A, A)$ is the total weights of edges inside group A. This measure quantifies how tightly the nodes of a group are connected to each other. Deriving the *Ncut* equation, we can define that

$$Ncut(A, B) = 2 - Nassoc(A, B).$$

It was shown by [17] and [16] that minimising the *Ncut* exactly is a NP-complete problem. However, [16] proposes a method to appoximate discrete

solutions efficiently for the *normalised cut* problem.

Indeed, the proposed algorithm starts by computing the associated similarities for graph $G = (V, E)$ for the image and store it in the matrices K and D.

Then, solve $(D - K)x = \lambda D x$ for the eigenvectors with the smallest eigenvalues. The way this equation is derived is explicitly shown in [16]. The eigenvector with the second smallest is used to bipartition the graph by finding the best splitting point to minimise $Ncut$. After that, it is possible to use the other eigenvectors to bipartition the graph even more. Once this is done, we can recursively apply the same algorithm on the segmented groups, stoping when the maximum number of groups is reached or the partitions are not good enough depending on an empirical threshold.

Edge separators and spectral rounding Introduced by [18], edge separators of a graph are created by iteratively reweighting edges until the graph disconnects into a certain number of groups. At each iteration, only a small number of eigenvectors are computed to reweight the edges. This spectral rounding method produces directly discrete solutions, and seems to compare nicely against the $Ncut$ approximation by producing more human-like results [19].

1.3   3D image processing

In general, [20] sets a very good overview of most variants and applications of spectral processing.

References

[1] Miroslav Fiedler. "Algebraic connectivity of graphs". eng. In: *Czechoslovak Mathematical Journal* 23.2 (1973), pp. 298–305. ISSN: 0011-4642. URL: `https://eudml.org/doc/12723` (visited on 10/26/2017).

[2] Fan R. K. Chung. *Spectral graph theory*. Vol. 92. CBMS Regional Conference Series in Mathematics. Published for the Conference Board of the Mathematical Sciences in Washington, DC; by the American Mathematical Society in Providence, RI, 1997, pp. xii+207. ISBN: 0-8218-0315-8.

[3] Drago M. Cvetkovi, Michael Doob, and Horst Sachs. *Spectra of graphs: theory and application*. en. Google-Books-ID: 4u7uAAAAMAAJ. Academic Press, 1980. ISBN: 978-0-12-195150-4.

[4] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. "A Review of Image Denoising Algorithms, with a New One". In: *SIAM Journal on Multiscale Modeling and Simulation* 4 (Jan. 2005). DOI: `10.1137/040616024`.

[5] K. Dabov et al. "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering". In: *IEEE Transactions on Image Processing* 16.8 (Aug. 2007), pp. 2080–2095. ISSN: 1057-7149. DOI: `10.1109/TIP.2007.901238`.

[6] Hossein Talebi and Peyman Milanfar. "Global Image Denoising". In: *IEEE Transactions on Image Processing* 23.2 (Feb. 2014), pp. 755–768. ISSN: 1057-7149, 1941-0042. DOI: `10.1109/TIP.2013.2293425`. URL: `http://ieeexplore.ieee.org/document/6678291/` (visited on 10/03/2017).

[7] Peyman Milanfar. *Non-conformist Image Processing with Graph Laplacian Operator*. 2016. URL: `https://www.pathlms.com/siam/courses/2426/sections/3234`.

[8] Hossein Talebi and Peyman Milanfar. "Asymptotic Performance of Global Denoising". en. In: *SIAM Journal on Imaging Sciences* 9.2 (Jan. 2016), pp. 665–683. ISSN: 1936-4954. DOI: `10.1137/15M1020708`. URL: `http://epubs.siam.org/doi/10.1137/15M1020708` (visited on 10/12/2017).

[9] Amin Kheradmand. "Graph-based image restoration". PhD thesis. University of California, Santa Cruz, 2016. URL: `http://search.proquest.com/openview/29f820ea2c8cd1f23d36a6a2bc4d3e7b/1?pq-origsite=gscholar&cbl=18750&diss=y` (visited on 10/12/2017).

[10] Wikipedia. *Difference of Gaussians — Wikipedia, The Free Encyclopedia*. `http://en.wikipedia.org/w/index.php?title=Difference\%20of\%20Gaussians&oldid=747011119`. [Online; accessed 19-October-2017]. 2017.

[11] Amin Kheradmand and Peyman Milanfar. "Non-Linear Structure-Aware Image Sharpening with Difference of Smoothing Operators". In: *Frontiers in ICT* 2 (Oct. 2015). DOI: `10.3389/fict.2015.00022`.

[12]   Peyman Milanfar. "Symmetrizing Smoothing Filters". en. In: *SIAM Journal on Imaging Sciences* 6.1 (Jan. 2013), pp. 263–284. ISSN: 1936-4954. DOI: `10.1137/120875843`. URL: `http://epubs.siam.org/doi/10.1137/120875843` (visited on 10/04/2017).

[13]   Z. Wu and R. Leahy. "An optimal graph theoretic approach to data clustering: theory and its application to image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.11 (Nov. 1993), pp. 1101–1113. ISSN: 0162-8828. DOI: `10.1109/34.244673`.

[14]   Francisco J. Estrada, Allan D. Jepson, and Chakra Chennubhotla. "Spectral Embedding and Min Cut for Image Segmentation." In: *Bmvc*. 2004, pp. 1–10.

[15]   Pedro F. Felzenszwalb and Daniel P. Huttenlocher. "Efficient Graph-Based Image Segmentation". en. In: *International Journal of Computer Vision* 59.2 (Sept. 2004), pp. 167–181. ISSN: 0920-5691, 1573-1405. DOI: `10.1023/B:VISI.0000022288.19776.77`. URL: `https://link.springer.com/article/10.1023/B:VISI.0000022288.19776.77` (visited on 10/17/2017).

[16]   jianbo shi and jitendra malik. "normalized cuts and image segmentation". In: *ieee transactions on pattern analysis and machine intelligence* 22.8 (2000), pp. 888–905. URL: `http://ieeexplore.ieee.org/abstract/document/868688/` (visited on 10/11/2017).

[17]   Christos H. Papadimitriou. "NP-completeness: A retrospective". en. In: *Automata, Languages and Programming*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, July 1997, pp. 2–6. ISBN: 978-3-540-63165-1 978-3-540-69194-5. DOI: `10.1007/3-540-63165-8_160`. URL: `https://link.springer.com/chapter/10.1007/3-540-63165-8_160` (visited on 10/18/2017).

[18]   David A. Tolliver and Gary L. Miller. "Graph partitioning by spectral rounding: Applications in image segmentation and clustering". In: *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 1*. IEEE Computer Society, 2006, pp. 1053–1060.

[19]   David A. Tolliver. *Spectral rounding and image segmentation*. Vol. 67. 11. 2006.

[20]   Hao Zhang, Oliver Van Kaick, and Ramsay Dyer. "Spectral mesh processing". In: *Computer graphics forum*. Vol. 29. Wiley Online Library, 2010, pp. 1865–1894.