Applications of Spectral Algorithms for Image Processing Tasks

Hui Han Chin
CMU-MCS
hchin@cmu.edu

Gary L. Miller
CMU-SCS
glmiller@cs.cmu.edu

Abstract

Spectral algorithms exploit information on the graph spectrum to gain computational speedups. Advances in spectral algorithms, such as spectral sparsifiers, fast symmetric diagonally dominant (SDD) system solvers of [KMP11] have created very powerful tools for the algorithmist. Most of the current fast technology are only available for weighted undirected graph and a good subset of undirected graph problems are image processing problems. They usually large in input size, ($\sim 10^7$ pixels is the data capture by a typical camera phone) and have good visuals as output. Also, there are many practical applications for image processing, such as medical imaging and robotics, that are beyond theoratical computer science. Improvements in image processing algorithms will definitely have a huge impact. Previous work in [MMP11] have shown that the popular total variation (TV) minimization framework used for image denoising, can be expressed in the electrical flows framework of [CKM+11]. Fast SDD solvers are then used in that framework, thereby leading to faster algorithms. In this thesis, we would study the applications of spectral algorithms on image processing tasks by expanding on the work of [MMP11]. We would

1. show the reduction of more image processing algorithms into the framework
2. provide implementation of the algorithms in code by building upon the fast SDD solver of [KMP10]
3. demostrate new applications of image processing algorithms in this framework on challenging dataset.

This approach is very flexible in describing different algorithms into the framework, thus allowing us to get more accurate results over traditional image processing algorithms by employing a mixture of algorithms simultanously. Also any improvements to the SDD solver, will translate instantly to speedup to the other algorithms of the framework. This approach of doing image processing has the advantage that different algorithms in the application pipeline to share common data structures and subrountines. Traditionally, image processing tasks are done imperatively. Our approach is theoratically interesting as it is a reformulation of the tasks as optimization problem over graphs.

1 Introduction

Image processing is any form of signal processing where the input is an image. An image maybe defined as a two-dimensional function $f(x, y)$, where x and y are *spatial* (plane) coordinates and $f : \mathbb{R}^2 \to \mathbb{R}^d$ is the intensity map of the image. When $d = 1$ we have grayscale images and when $d = 3$ we have color images in the RGB channel. Higher dimensions of d corresponds to hyper spectral imaging which are very common in medical imaging and remote sensing. Thus an image can be thought of as a weighted finite graph with potentials on the vertices. Often, a vertex in the graph would be used to represent a pixel in the image.

The two main limitations in image capturing are **blur** and **noise**. Blur is an inherent property of image acquisition systems as the continuous stream of light is only sampled at finite elements of the sensor and is thus subjected to the Shannon-Nyquist sampling conditions. Also, blur can be a result of unwanted motion during the image capturing process. Noise is due to fluctuations in the sensor readings of the image acquisition system. Even when the incoming light is a constant source, one can expect approximately $\sqrt{n}$ fluctuations for every n photons by the central limit theorem. This is often due to random emission of

photons from the heat of sensor. Such background noise have been widely studied and they are commonly as Additive Gaussian White Noise (AWGN). Also, the environment might introduce a consistent source of noise which might not conform to the AWGN model.

Image processing are often low processes which used in more complex applications such as computer vision, machine learning and robotics. For example, denoising and clustering are often done as preprocessing of object recognition in computer vision. Since image processing occurs early in the application pipeline, it is of great importance that it is done accurately and quickly. Two fundamental problems in image processing are 1) restoration of images that are corrupted by noise or large artifacts and 2) segmentation of images into regions that are "similar" with respect to certain measures such as visual coherence.

1.1 Previous Work on Image Denoising

Image Denoising is a very well studied problem due to its wide spread applications and many textbooks, such as [GW01], have been written on the subject. Most image processing techniques have their origins in Digital Signal Processing (DSP) thus it is no surprise that some techniques such as Gaussian Low Pass filtering, Median filtering are common to both fields. However, such cannonical techniques are only useful when the noise structure is known completely beforehand. Also they do not consider the image structure. Modern techniques such as those in cited in survey papers such as [BCM04] provides estimates for the noise structure while considering the image structure. Techniques such as collaberative filtering [DFKE09] and non local means has emerge as preferred denoising algorithms as they often give visually appealing images. The theme in those techniques is that the noise structure are estimated by sampling the image at smaller patches. However such techniques are local and they do not take into account of global properties of the image. One image processing model that was of particular interest to us is the Rudin, Osher , Fatemi (ROF) model in [ROF92]. It is a model that solve the Total Variation inverse problem in [CS05] and proven to be effective in practice as it encompasses both local and global properties of an image.Also, it is more effective than most techniques at preserving sharp features such as edges in denoising. Also it was shown in [MMP11] that the ROF model can be expressed as a SDD system and thus we can take advantage of fast SDD solver technology such as [KMP10].

1.2 Total Variation Minimization

The most general form of a total variation minimization problem is given an image with intensities $\mathbf{s}$, try to recover an image $\mathbf{x}$ that minimizes the following:

$$FID(\mathbf{x} - \mathbf{s}) + TV(\mathbf{x})$$

Where FID is the fidelity term that measures how much the image is altered and $TV(\mathbf{x})$ ensures the smoothness of the image [DS06]. Although the underlying model of the image is continuous, the pixel nature of the input s means that TV also needs to be computed discretely. This is often done by setting it to be a function of differences of neighboring pixels, which can conveniently be represented as edges of a graph $G = (V, E)$. It's worth noting that if we drop the fidelity term, then the best $\mathbf{x}$ returned is a uniform image; while if we drop the TV term, the best $\mathbf{x}$ returned is identical to the input image that includes the noise. The setting where FID is L_2^2 and TV is L_1, known as anisotropic total variation is especially appealing due to the key property that minimizing L_1 functions results in sharp boundaries [ROF92]. An extra multiplicative term of λ is used to leverage the fidelity and smoothness terms against each other, giving the following formulation objective

$$||\mathbf{x} - \mathbf{s}||_2^2 + \lambda \sum_{(u,v)\in E} |\mathbf{x}_u - \mathbf{x}_v|$$

In [MMP11], a more general version of TV denoising on the ROF model was stated.

Definition 1.1 *The* **generalized isotropic TV minimization problem** *is defined as follows:*

Input: A weighted undirected graph $G = (V, E, \boldsymbol{cost}, \mathcal{S})$ *with non-negative edge weights cost, a partition of the edges* E*:* $\mathcal{S} = S_1, \ldots, S_k$*, and potentials* $s : W \subset V \to \mathbb{R}$

Output:

$$\min_{\boldsymbol{y}} \mathcal{TV}_G(\boldsymbol{y}) \quad \textit{subject} \quad \boldsymbol{y} \restriction W = \boldsymbol{s}$$

Where:

$$\mathcal{TV}_G(\boldsymbol{y}) = \sum_{1 \le i \le k} \sqrt{\sum_{(u,v) \in S_i} \boldsymbol{cost}_{uw} (\boldsymbol{y}_u - \boldsymbol{y}_v)^2}$$

Each subset of edges $S_1, \ldots, S_k$ a **cluster**. Define the **Root Difference Squares** of a cluster to be $\mathbf{RDS}(\mathbf{y}, i) = \sqrt{\sum_{(u,v) \in S_i} \mathbf{cost}_{uw} (\mathbf{y}_u - \mathbf{y}_v)^2}$ then $\mathcal{TV}_G(\mathbf{y}) = \sum_{1 \le i \le k} \mathbf{RDS}(\mathbf{y}, i)$.

Also they have shown common formulations of TV still reduce to this version under suitable searches of parameters. It is important to note that the above version of the formulation is for the unit weighted graph, shown for notation convinence. In their paper, they gave an analysis of weighted version.

1.3 Our Contributions

Efficient image denoising and segmentation is still a valid challenge. They stand at the crossing of functional analysis and statistics. Most state of the art algorithms have outstanding performance when the image model corresponds to the algorithms' assumption but fail in general. Images are often composites of smaller images of different objects, often known as patches, and each patch might have a different image model. Even though it is possible to apply specialized algorithms on individual patches, often this result in decoherent patches where global features, such as a general illumination, are lost due to localization.

The starting point of this paper is an observation that the interpretation of the ROF model can be easily expanded to a more general framework. We can view the fidelity term as our prior belief or assumptions on the image input. A common assumption would be the image being corrupted with gaussian or poisson noise and in the ROF case, a gaussian noise was assumed. Similarly, the smoothness term is our prior belief or assumptions on the output of image processing. The term $X - S$ is often called the noise image as it encodes information of the noise structure.

$$FID(\mathbf{x} - \mathbf{s}) + TV(\mathbf{x}) \Rightarrow \text{Noise Assumption} + \text{Image Assumption}$$

It was noted in [MMP11] that such formulation has not been fully exploited in previous works is that the underlying linear systems that are needed to solve are symmetric diagonally dominant. This, in turn, allows one to leverage the significant progress that has been made over the last decade in solving such systems [BH03, ST06, KM07, KMP10, KMP11], leading up to algorithms that run in nearly-linear time, depending on the application.

We would expand on the framework describe in [MMP11] and [CKM+11] to incorperate more algorithms show novel applications by using a mixture of these algorithms. Our technical contributions are as following:

1. show the reduction of more image processing algorithms into the framework
2. provide implementation of the algorithms in code by building upon the fast SDD solver of [KM07]
3. provide heuristic for speeding up the algorithms.
4. demostrate new applications of image processing algorithms in this framework on challenging dataset.

Our framework allows a mixture of image processing algoritms to be employed simulatenously on an image while striking a balance between all the algorithms. Our main technical contribution allows us to

overcome the shortcoming of traditional image processing algorithms and show that we can outperform most of them in both speed and accuracy. Even though the traditional TV approach is state of the art, we are able to improve on it by solve the problems of false artifact creation that is associated with the traditional TV approach. Also, the flexibility of our framework allows many different variant of image processing algorithms, giving us novel applications.

The organization of this thesis is as follows: in Section 2 we outline our formulation of the problem. In Section 3 we show the reduction of known image processing technique into our framework. In Section 4 we outline the algorithms that are build upon the SDD solver that are used in our work. In Section 5 we demostrate applications of our framework on image processing task. In Section 6 we discuss the potential expansion of our framework.

2 Formulation and Notations

2.1 Notations

As in the Definition 1.1 we use $G = (V, E, P, W)$ to denote an abstract graph, but one may think of it as formed by the pixels and their neighboring relations. Throughout the thesis, upper letters would be used to denote sets, while lower case would be used to denote the values which the member of the set adopt at some instance. Let $S \subset V$ be the subset of vertices with fixed potentials. This subset represents the sensor input. Let $X \subset V$ be the subset of vertices with variable potentials. This subset represents the output that we obtain. Note that S and X partitions V. To describe the connectivity on the graph, we will have the following definitions. For each neighboring pair of vertices/pixels, we assign one of them arbitrarily as the head/tail of the edge/neighboring relation, and Γ to denote its edge-vertex incidence matrix:

$$\Gamma_{e,u} = \begin{cases} -1 & \text{if u is the head of e} \\ 1 & \text{if u is the tail of e} \\ 0 & \text{otherwise} \end{cases}$$

It's worth noting that our algorithm does not rely on the underlying structure of the neighboring relations between pixels. This makes the algorithm readily applicable to 3-D images or non-local models involving the addition of edges across the image . However, when the neighborhoods are those of a 2-D image, a saving by a $\log n$ factor can be obtained by using the optimal solver algorithm of [KM07]. Now observe that the edge set E is related to the vertex set V by

$$\Gamma V = E$$

Let a subset of the edge set E be called a cluster and P be the partition on the powerset of E. Elements of P denotes the clusters of G. let W be the cluster weight function $W : P \to \mathbb{R}$. Note that is we only consider 1-subset of E, then we get the traditional notion of a weighted edge. Let w_{ij} be the weight on the edge ij. Let the graph laplacian matrix, $\mathbf{L}$ of G be defined as

$$\mathbf{L}_{i,j} = \begin{cases} \sum_j w_{ij} \text{ if } i = j \\ -w_{ij} \text{ otherwise} \end{cases}$$

and we can see that $\mathbf{L}$ is a SDD linear system. Also the following formulae are known.

$$\mathbf{L} = \Gamma^T W \Gamma$$

Let the graph at t iteration be G^t and let F be reweighting scheme such that $F : G^t \to W^{t+1}$. F takes in a graph and computes the cluster weights in the next iteration. The use of F will be more apparent when we discuss the $MultiplicativeWeights$ framework in the next section.

2.2 Problem Formulation

Definition 2.1 *The* **Image Graph Optimization problem** *is defined as follows:*

Input: A weighted undirected graph $G = (V, E, P, W)$ *with* $S \cup X = V$

Output:

$$\min_X \mathcal{IGO}_G(X) \quad \textit{subject} \quad V \restriction S = s$$

Where:

$$\mathcal{IGO}_G(X) = \sum_{1 \le i \le k} w_i \sqrt{\sum_{(u,v) \in P_i} w_{uw}(x_u - x_v)^2}$$

This problem in general is hard to solve. However, the crucial observation is that we need have apprioriate weighting function F that sets the weights w_{uw} such the the overall function is convex. Then we can use a quadratic programming to solve for the objective function. Two such objective functions are the $L1$ and $L2$ norms.

3 Reduction of Other TV Variants to Our Formulation

Although our formulation of $\mathcal{TV}$ as a sum of L_2 objectives differs syntactically from some common formulations, we show below that the more common formulation involving a L_2-squared fidelity term can be reduced to it using 2 iterations of ternary search. Most other formulations differs from our formulation in the fidelity term, but more commonly have L_1 smoothness terms as well. Since the anisotropic smoothness term is a special case of the isotropic one, our discussion of the variations will assume anisotropic objectives.

3.1 L_2^2 fidelity term

The most common form of the total variation objective in literature is one with L_2^2 fidelity term.

$$\min_{\mathbf{x}} ||\mathbf{x} - \mathbf{s}||_2^2 + \sum_{1 \le i \le k} \sqrt{\sum_{uv \in S_i} (\mathbf{x}_u - \mathbf{x}_v)^2}$$

Since we are treating the $\mathbf{s}$ as variables with fixed values, we can express $||\mathbf{x} - \mathbf{s}||_2^2$ as a quadratic form involving $\mathbf{g}$ which we'll denote as $\mathbf{I}_0$ ($\mathbf{g} = \Gamma_1\mathbf{x} + \Gamma_2\mathbf{s}$. Using $\mathbf{I}_1 \ldots \mathbf{I}_k$ as indicator diagonal matrices for $S_1 \ldots S_k$ gives:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{g}} \quad & \mathbf{g}^T\mathbf{I}_0\mathbf{g} + \sum_{i=1}^{k} \sqrt{\mathbf{g}^T\mathbf{I}_i\mathbf{g}} \\ & \mathbf{g} = \Gamma_1\mathbf{x} + \Gamma_2\mathbf{s} \end{aligned}$$

Instead of putting $\mathbf{g}^T\mathbf{I}_0\mathbf{g}$ in the objective, we can establish its value separately by guessing it as a constraint. Since the L^2 is convex in L, the following optimization problem is convex in L as well:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{g}} \quad & \mathbf{g}^T\mathbf{I}_0\mathbf{g} + \sum_{i=1}^{k} \sqrt{\mathbf{g}^T\mathbf{I}_i\mathbf{g}} \\ & \mathbf{g}^T\mathbf{I}_0\mathbf{g} \le L^2 \\ & \mathbf{g} = \Gamma_1\mathbf{x} + \Gamma_2\mathbf{s} \end{aligned}$$

Since L^2 is convex, ternary searching on L would allow us to find the optimum solution by solving $O(\log n)$ instances of the above problem. Taking square root of both sides of the $\mathbf{g}^T\mathbf{I}_0\mathbf{g} \le L^2$ condition and making the Lagrangian relaxation gives:

$$\min_{\mathbf{x},\mathbf{g}=\Gamma_1\mathbf{x}+\Gamma_2\mathbf{s}} \max_{\lambda\geq 0} \sum_{i=1}^{k} \sqrt{\mathbf{g}^T\mathbf{I}_i\mathbf{g}} + \lambda(\sqrt{\mathbf{g}^T\mathbf{I}_0\mathbf{g}} - L)$$

Which by the min-max theorem is equivalent to:

$$\max_{\lambda\geq 0} -\lambda L + \left(\min_{\mathbf{x},\mathbf{g}=\Gamma_1\mathbf{x}+\Gamma_2\mathbf{s}} \sum_{i=1}^{k} \sqrt{\mathbf{g}^T\mathbf{I}_i\mathbf{g}} + \lambda\sqrt{\mathbf{g}^T\mathbf{I}_0\mathbf{g}} \right)$$

The term being minimized is identical to our formulation and its objective is convex in λ when $\lambda \geq 0$. Since $-\lambda L$ is linear, their sum is convex and another ternary search on λ suffices to optimize the overall objective.

3.2 L_1 fidelity term

Another common objective function to minimize is where the fidelity term is also under L_1 norm. In this case the objective function becomes:

$$||\mathbf{x} - \mathbf{s}||_1 + \sum_i \sum_{1\leq i\leq k} \sqrt{\sum_{(u,v)\in S_i} (\mathbf{x}_u - \mathbf{x}_v)^2}$$

This can be rewritten as a special case of $\mathcal{TV}$ as:

$$\sum_u \sqrt{(\mathbf{x}_u - \mathbf{x}_u)^2} + \sum_i \sum_{1\leq i\leq k} \sqrt{\sum_{(u,v)\in S_i} (\mathbf{x}_u - \mathbf{x}_v)^2}$$

In which case invoking Theorem 5.4 of [MMP11] gives an algorithm that produces a solution with objective at most $(1+\epsilon)$ times the optimal in time $\tilde{O}(m^{4/3}\text{poly}(\epsilon^{-1}))$.

3.3 Reduction to local filters

Once we have establish the $L1, L2$ norms in our framework, we can apply the results such as [LM11] to incorperate a large class of local filters into the extended ROF framework.

4 Features of our Fast Graph Optimization toolbox

4.1 Core routine

Building on the fast SDD solver of [KM07]

1. Multiplicative Weights Solver
2. K-Eigensystem Solver

4.2 Multiplicative Weights Minimizer

The core algorithm for IGO minimization is shown in Figure 1. The algorithm extends the ideas used in the cut algorithm from [CKM+11] to the case of non-linear objective functions. It assigns a weight $\mathbf{w}_i$ for each cluster of edges, and iteratively reweights them in order to identify a good set of cuts. In each iteration, it assigns resistances to the edges of the graph based on those weights and then computes an almost-optimal vertex potential/electrical using those resistive values.

The guarantees for this algorithm are proven in [MMP11]

Algorithm 1 Algorithm for finding vertex potentials

MultiplicativeWeights

Input: Weighted graph $G = (V, E, S_1, \ldots, S_k)$ with corresponding edge-vertex incidence matrix $\Gamma = [\Gamma_1 | \Gamma_2]$. Fixed values $\mathbf{s}$ for vertices. Target objective value F. Width parameter ρ. Error bound ϵ.
Output: Either vectors $\mathbf{y}$ such that $\mathcal{TV}_G(\mathbf{y}) \leq (1 + 10\epsilon)F$. Or **fail** indicating that $F < F^*$.

Initialize $\mathbf{w}_i^0 = 1$ for all $1 \leq i \leq k$
$N \leftarrow 10\rho \log n \epsilon^{-2}$
$\delta \leftarrow \frac{\epsilon^2}{k}$
for $t = 1 \ldots N$ **do**
  for $i = 1 \ldots k$ **do**
    for $e \in S_i$ **do**
      $\mathbf{r}_e \leftarrow w_i^{t-1}$
    end for
  end for
  $(\tilde{\mathbf{x}}, \tilde{\mathbf{f}}) = \textsc{Oracle}(G = (V, E), \mathbf{s}, \mathbf{r}, \delta)$
  $\mu^{t-1} \leftarrow \sum_i \mathbf{w}_i^{t-1}$
  if $\mathcal{P}_{potential}(\mathbf{r}, \tilde{\mathbf{y}}) \leq \frac{F^2}{\sum_i \mathbf{w}_i^{t-1}}$ **then**
    return $\tilde{\mathbf{y}}$
  else
    for $1 \leq i \leq k$ **do**
      $\mathbf{w}_i^t \leftarrow \mathbf{w}_i^{t-1}(1 + \frac{\epsilon}{\rho}\mathbf{cong}(\tilde{\mathbf{f}}^t, i)) + \frac{\epsilon^2}{k\rho}\mu^{t-1}$
    end for
  end if
end for
return **fail**

Multiplicative weights algorithm for finding a set of vertex potentials with a small objective. The algorithm maintains a set of weights, one per cluster S_i starting at 1. In each iteration it computes an approximate vertex potential/ electrical flow with flow value F using resistive values on edges corresponding to the weights of the sets. The weights of each cluster is then updated multiplicatively using the total energy of the edges in that cluster.

Algorithm 2 Algorithm for finding the second eigenvector and eigenvalues

SECONDEIGENSYSTEMSOLVER

Input: Sparse matrix A that is graphable. Sparse matrix B which is the target subspace. Integer k. Error bound ϵ.

Output: the second pair of (λ_2, x_2) such that $Ax_2 = \lambda_2 Bx_2$.

Initialize x to a random column vector.
$b_0 \leftarrow B.\mathbf{1}$
$q_0 \leftarrow 0$
$q_1 \leftarrow 1$
while $|q_i - q_{i-1}| > \epsilon$ **do**
    $\alpha = \frac{X^T b}{b^T b}$
    $X = X - \alpha b$
    $q_i \leftarrow \frac{x^T Ax}{x^T Bx}$
    SDDSOLVE(A, BX)
end while
$\alpha = \frac{X^T b}{b^T b}$
$X = X - \alpha b$
$\lambda \leftarrow \frac{x^T Ax}{x^T Bx}$
return (λ, X)

SecondEigensystem algorithm for finding the second eigenpair of a graphable matrix A. Since A is graphable, the first eigenvector is the all ones vector $\mathbf{1}$. The algorithm solves for the second eigenvector by inverse powering. A random guess for an eigenvector, x is initialized. At every iteration, the algorithm solves for $Lx = x$ and reproject it into the subspace of $B.\mathbf{1}$. Once we found a x whose reprojection distance is 0, we found the eigenvector. From that eigenvector, it computes the associated eigenvalue. To find the subsequent eigenvectors, it project it into the subspace that is spanned by the previous eigenvectors found. This process is known as deflation and is handled by a top level rountine.

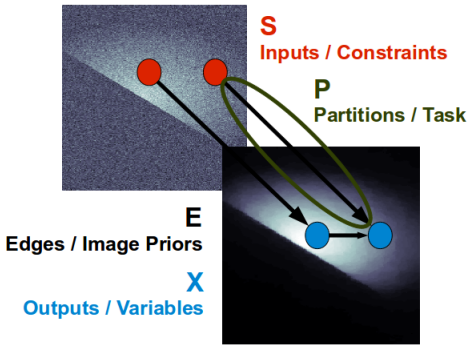
4.3 Second Eigensystem solver

This solver is mainly used by the spectral segmenting algorimth. The second eigenvector of a graph corresponds to the fundamental mode of vabiration which gives the first major cut of the graph. We can then iteratively cut up the graph into smaller pieces by applying the algorithm.

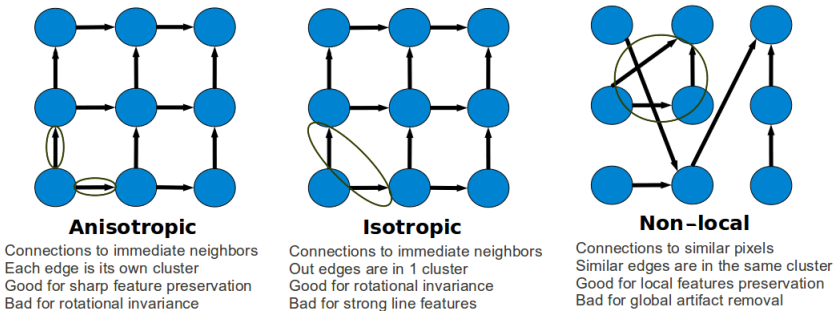
Building upon this 2 routines, our toolbox implements the following core routines.

1. The IGO denoiser
2. Spectral Segmentor

The IGO denoiser is a direct application for the Multiplicative weight solver. The reweighting functions correspond to the multiplicative weight scheme. We view images now as graphs as per the formulation in



It is important to note that all routines share the common data structure, the edge-vertex incidence matrix Γ and we give an efficient implementation of that and the reweighting functions using a sparse representation with vectorize operations. Also our toolbox supports the following connectivity structures.

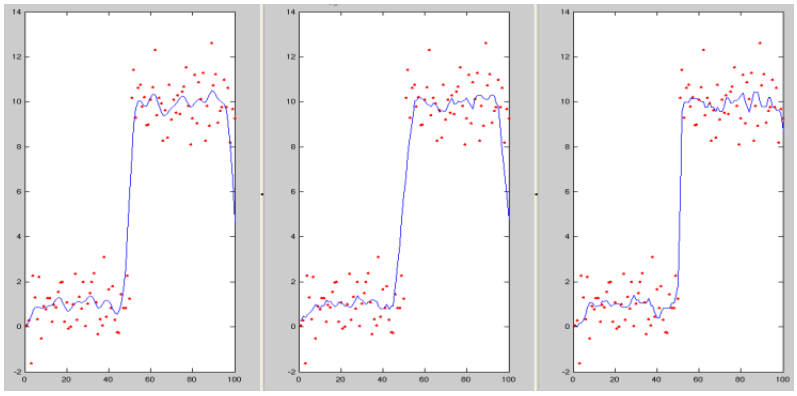


Also, since this is build upon a very fast SDD solver, we have about 60 times speed up when compared to the state of the art implementations of the traditional algorithms. Since the solver uses only $O(n \log n)$ memory as compared to the $O(n^2)$ implmentation of matlab, we are able to tackle much larger problem sets.

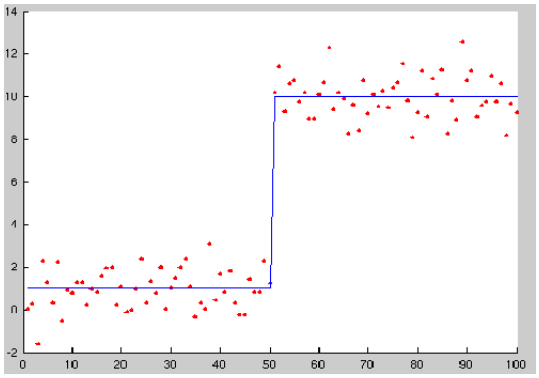
5 Applications

5.1 Signal Processing

In classical signal filtering approaches, prior belief of the noise structure is encoded in the choice of the filter. However, classical filters are unable to express mixtures of priors due to the lack of a framework. For example, given a step signal corrupted by gaussian noise, the blue lines would be what typical signal processing filters would recover.

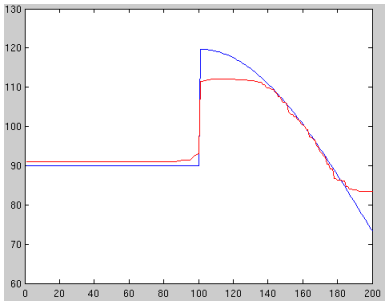


Using the ROF model as per [ROF92], we can encode a Gaussian belief on the the noise structure by using a $L2$ norm on the clusters running across the $X - S$ and encode a laplacian belief on the edges within X. We are able to achieve the following result.

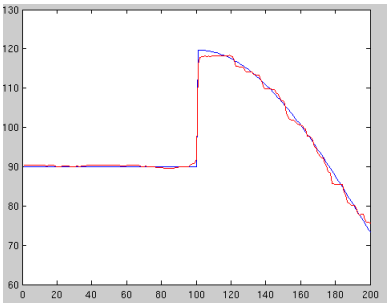


Note that the solution recovered simulatenously statisfies both beliefs on the noise and signal structure in order to achieve almost 100% signal recovery. Also we are able to avoid the traditional staircase artifact of the TV model by using the clipped loss functions.

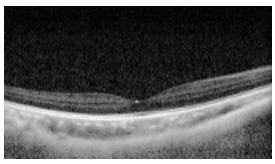
Typical TV implementation



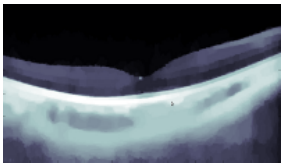
Our TV implementation



5.2 Image Denoising



Noisy optical coherence tomography image of retina



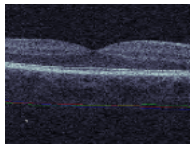
denoise image that preserve key features.

In this application, using $L1/L2$ total variation minimization algorithms we are able to denoise images obtain from optical coherence imaging of the retina. Using a combination of the $L1/L2$ reweighting schemes, remove noise while preserving the sharpness between the nerve fiber layers. The nerve fiber layers are of clinical importance as anomalities often are indication of diseases.
Also we are able to outperform state of the art methods on a standard dataset.

Noisy Leena	GoldsteinOsher 09	Iso TV Micchelli 11	RDS TV, Peng 11	Clipped TV , Chin 12
				
22.5E6	13.0E6	14.6E6	3.89E6	**3.26E6**

The last line indicts the total pix error from the ground truth. The input image is 512 by 512.

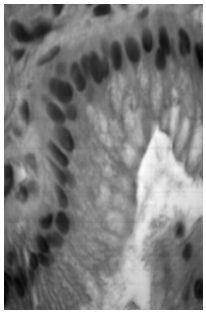
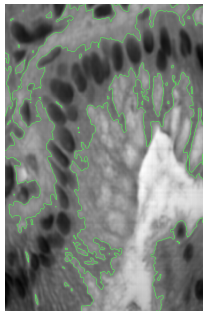
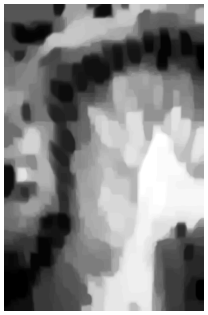
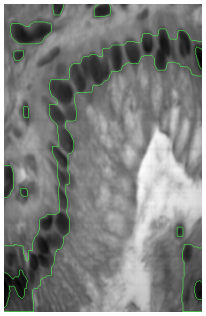
5.3 Iterative Spectral Segmentation




OCT image of retina top 4 cuts of the OCT image

Using the spectral cut algorithm, we are able to obtain segmentations of very thin nerve fiber layers. This approach is superior over edge based segmentation techniques as we can incorperate different priors on the segmentation. In this example, we incorperate the prior that nerve fibers are thin horizontal strips. The key advantage is that we are able to segment and denoise in an iterative manner thus give us better segmentation. Segmentation is a downstream process that is very sensitive process. However, if we denoise aggressively, we loss fine details, thus our toolbox gives us precise control over the 2 process.

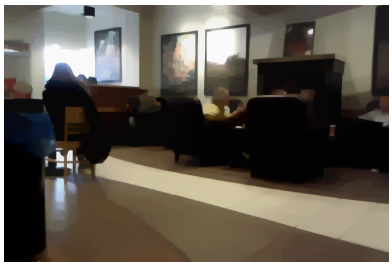
5.4 Segmentation Guide

Original lung tissues Naive segmentation using MATLAB Aggressive denoising Same segmentation algorithm

If specialized segmentation algorithm are preferred, we can use the denoising result as a guide for the downstream segmentation algorithm. Since denoised result remove local noise while maintaining global features such as long edges, it has a tendency of boosting the accuracy of the downstream algorithms. In this example, MATLAB's builtin segmentation algorithm is used to segment the nucleus (the black dots) of lung cell in a lung scan. Since the algorithm uses only local morphological operators, it is naive and and not robust. However, by doing denoising and using the denoise image as a guide, we can get closer segmentations.

5.5 Color Correction

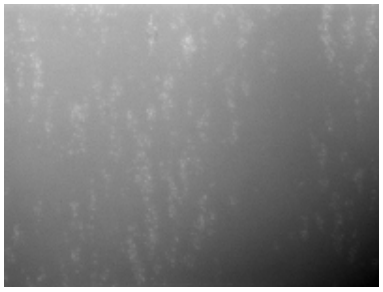
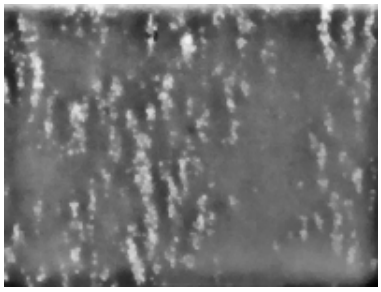
Color image Globally stable features

By setting the apprioriate ratio between the reweighting schemes of the input layer and the output layer, we are able compute globally stable features of images such a global illumination. These are features that likely to remain unchange under local transformation and they appear different scale spaces. Previous approaches that compute such features uses gaussian filtering which is an $L2$ optimization essentially. Our approach allows the preservation of sharp features such as edges. Such features are of interest to applications such as SIFT and object tracking for the robotics and vision coummunity.


newpaper scan with visual artifect corrected image

In addition to grayscale images, we can treat color images as graphs with vertices in 3 dimensional space, where each pixel is a combination of the RGB channel. We can reformulate the denoising problem as color correction if there are visual artifect that are present in a single color channel. This would have applications in optical character recognition tasks as the characters are preserve while large artifects, such as the yellowing of newspaper, are corrected.

5.6 Signal Boosting

blood platelet with visual artifects and occlusions boosted image

Combing all the approaches, we can boost weak signals by iteratively removing global features while accentuating local features. However, the algorithm is able to distinguish noise from true local features, such as blood platelets in the above example.

6 Discussions and Conclusions

6.1 Future Work

The following areas are scheduled for more through analysis.

1. Numerical stability of clipped loss functions
2. An/Isotropic connectivity and rotational invariance in images

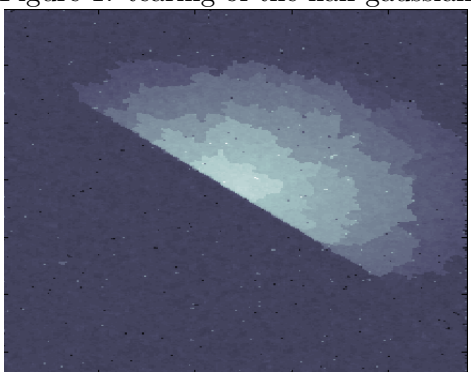
6.1.1 Numerical stability of clipped loss functions

The most pressing issue that needs to be studied is the numerically stablity of the clipped loss function. The function is non-convex and thus creates numerical stability issues for the solver as it violates the assumptions of the objective function. This issue have been observe in practice where the solver stagnates. However the issue can be remedied by taking smaller step size at each solver iterations such that we are in approximately convex regions of the clipped loss function. Thus the relationship of step size to the degree of clipping used needs to be carefully studied to make this approach robust.

6.1.2 An/Isotropic connectivity and rotational invariance in images

When isotropic connectivity was proposed in [ROF92], the author had in mind that majority of images exhibit small degree of rotational invariance, meaning that images do look the same when rotated slightly. In [CS05], the anisotropic connectivity was considered as it was easier to solve and it has the property of maintaining sharp edges. However, the anisotropic connectivity has a tendency of introducing the staircase artifact. Through our experiments, we discovered that isotropic connectivity together with the clipped loss functions lead to phenomena where the image "tears" when the rotational invariance assumption if violated. For example, in the case of the half gaussian, the image is not rotationally invariant due to the dominant

Figure 1: tearing of the half gaussian



diagonal edge. The tearing is increased by the clipped function and thus the clipped loss function is best used with the anisotropic connectivity. Since it is not possible to gurantee that the input image is rotationally invariant, we propose to reexpress the connectivity as $\sqrt{\alpha dx^2 + (1-\alpha)dy^2}$. Observe that when $\alpha = 0$ we get back the anisotropic connectivity. Just as the weighting functions allows us to vary between the $L1$ and $L2$ norms, we hope that this formulation would allow finer control over the image connectivity and thus give better results.

6.2 Conclusions

We have presented a theoratically interesting way of reformulating imperative image processing as optimization problem over graphs. This have also been highlighted in many other papers and this reformulation allows greater flexibility and control over traditional imperative image processing methods. However this

formulation have not caught on as it has huge memory overhead and a huge runtime in an naive implementation. Even though we are not the first group to realize formulation, we are the first in incorperating the fast SDD solver of [KM07] into the framework, thus making it a viable method. Also we have presented an extension to the framework by incorperating the clipped loss function. This allows us to avoid the staircase artifact issues that is often associated with the TV denoising method. Even though the clipped loss function due lead to numerical instability issues, we observe that in practise, it can does not occur frequently and can be easily avoided by using a smaller step size. Our implementation of the extended ROF framework is not only faster, but also has a smaller memory overhead as compared to the state of the art. This allows us to handle larger datasets than the state of the art. Also by exploiting our speed, we are able to create novel applications such as weak signal boosting and iterative segmentation. We believe that given the flexibility of the framework, we can extend the techniques to problems beyond image processing such as signal processing.

References

[BCM04] Antoni Buades, Bartomeu Coll, and Jean Michel Morel. On image denoising methods. Technical report, Technical Note, CMLA (Centre de Mathematiques et de Leurs Applications, 2004. 1.1

[BH03] Erik G. Boman and Bruce Hendrickson. Support theory for preconditioning. *SIAM J. Matrix Anal. Appl.*, 25(3):694–717, 2003. 1.3

[CKM+11] Paul Christiano, Jonathan A. Kelner, Aleksander Mądry, Daniel Spielman, and Shang-Hua Teng. Electrical Flows, Laplacian Systems, and Faster Approximation of Maximum Flow in Undirected Graphs. In *Proceedings of the* 43^{rd} *ACM Symposium on Theory of Computing (STOC)*, 2011. (document), 1.3, 4.2

[CS05] Tony Chan and Jianhong Shen. *Image Processing And Analysis: Variational, Pde, Wavelet, And Stochastic Methods*. SIAM, 2005. 1.1, 6.1.2

[DFKE09] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. BM3D Image Denoising with Shape-Adaptive Principal Component Analysis. In Rémi Gribonval, editor, *SPARS'09 - Signal Processing with Adaptive Sparse Structured Representations*, Saint Malo, France, 2009. Inria Rennes - Bretagne Atlantique. 1.1

[DS06] Jérôme Darbon and Marc Sigelle. Image restoration with discrete constrained total variation part i: Fast and exact optimization. *Journal of Mathematical Imaging and Vision*, 26(3):261–276, 2006. 1.2

[GW01] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001. 1.1

[KM07] Ioannis Koutis and Gary L. Miller. A linear work, $O(n^{1/6})$ time, parallel algorithm for solving planar Laplacians. In *Proc. 18th ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*, 2007. 1.3, 2, 2.1, 4.1, 6.2

[KMP10] Ioannis Koutis, Gary L. Miller, and Richard Peng. Approaching optimality for solving SDD systems. *CoRR*, abs/1003.2958, 2010. 2, 1.1, 1.3

[KMP11] Ioannis Koutis, Gary L. Miller, and Richard Peng. Solving sdd linear systems in time $\tilde{O}(m \log n \log(1/\epsilon))$. *CoRR*, abs/1102.4842, 2011. (document), 1.3

[LM11] Cécile Louchet and Lionel Moisan. Total variation as a local filter. *SIAM J. Img. Sci.*, 4(2):651–694, June 2011. 3.3

[MMP11] Aleksander Madry, Gary L. Miller, and Richard Peng. Electrical flow algorithms for total variation minimization. *CoRR*, abs/1110.1358, 2011. (document), 1.1, 1.2, 1.3, 3.2, 4.2

[ROF92] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithm. *Physica D*, 1(60):259–268, 1992. 1.1, 1.2, 5.1, 6.1.2

[ST06] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *CoRR*, abs/cs/0607105, 2006. 1.3