Enforcing Point-wise Priors on Binary Segmentation

Feng Li
http://www.fengl.org/
Light
Palo Alo, CA 94301, USA

Fatih Porikli
http://www.porikli.com/
Australian National University / NICTA
Canberra ACT 0200, Australia

Abstract

Non-negative point-wise priors such as saliency map, defocus field, foreground mask, object location window, and user given seeds, appear in many fundamental computer vision problems. These priors come in the form of confidence or probability values, and they are often incomplete, irregular, and noisy, which eventually makes the labelling task a challenge. Our goal is to extract image regions that are aligned on the object boundaries and also in accordance with the given point-wise priors. To this end, we define a graph Laplacian spectrum based cost function and embed it into a minimization framework. For a comprehensive understanding, we analyze five alternative formulations, and demonstrate that the robust function version produces consistently superior results.

1 Introduction

Binary segmentation, e.g. finding an object area in a given image, partitioning a foreground region in a scene, and detecting an organ surface in a volumetric medical data, is a fundamental computer vision task. In this paper, we analyze the seminal work in binary segmentation including graph-cuts [3, 15] , random walk [7], alpha matting [11, 12], and spectral clustering [14], and present a unifying framework that shows these methods share similarities that can be modified to synthesize a specific formulation for each of them.

In addition, we describe a novel graph Laplacian spectrum constraint to impose structure and point-wise constraints on the segmentation task. We present five alternative formulations including projection onto null-space, convex function with ℓ_2-norm, convex function with ℓ_1-norm, sparse decomposition, and iterative robust function. We demonstrate that the robust solution generates superior results for several preferred applications on 2D images. Our formulation can be extended easily to any graph bipartitioning problems in higher dimensional spaces such as clustering vector data.

Let $\mathbf{x}$ be the binary segmentation result that we seek. We are given with a structure $\mathbf{y}$ that we want to impose on the segmentation, and a prior $\mathbf{x}^\star$ (or an initial estimate) that guides the segmentation process, as illustrated in Fig. 1. For this binary segmentation problem, $\mathbf{y}$ can be the original image in vector form, $\mathbf{x}^\star$ be the likelihood weights (or confidence map) indicating an image pixel belonging to target object, and $\mathbf{x}$ be the set of labels we aim to detect.

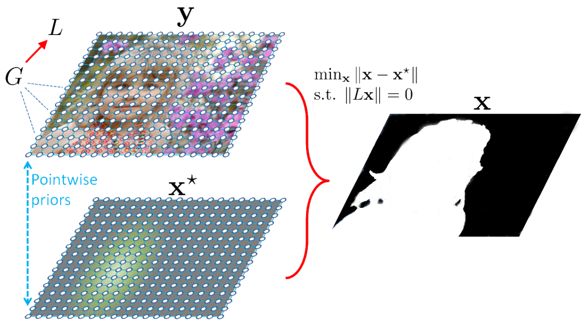


Figure 1: Binary segmentation by Laplacian spectrum constraint. We compute a graph Laplacian matrix L from graph G for the input structure y to regularize the estimation of $\mathbf{x}$ from $\mathbf{x}^\star$.

We consider $\mathbf{y}$ is represented with a graph G. This structure imposing graph is built by assigning each point in $\mathbf{y}$ as a vertex and connecting these vertices via weighted edges within a N-connectivity. In other words, each vertex is an image pixel and each edge is the affinity value of two pixels in a patch $\mathcal{N}$ containing $N+1$ pixels including the center. Therefore, G is a highly sparse and almost an N regular graph except on the boundary vertices, which have less than N neighbors. For a 2D image $\mathbf{y}$ of size $\sqrt{n}\times\sqrt{n}$, G will have n vertices. Different connectivity and weighting schemes will generate different weighted graphs. The graph Laplacian L is a positive-semidefinite matrix representation of G. For properties of the graph Laplacian, please refer to [5].

2 Related Work

Segmentation is one of the extensively studied areas in computer vision. Providing an in-depth analysis of the existing work is outside the limited scope of this paper. Here, we provide a summary of a few relevant studies.

Pioneering work [15] treats image segmentation as a graph partitioning problem and proposes a normalized cut criterion measures both the total dissimilarity between the different groups as well as the total similarity within the groups. Random walk [7] is a seeded segmentation algorithm that determines the probability that a random walker starting at each unlabeled pixel will first reach one of the prelabeled pixels by solving a closed form equation using graph Laplacian where weights $A(i,j)=\exp(-\theta(g_i-g_j)^2)$, where θ is the global scaling factor, in contrast to local scalings in [11, 14], and g_i is image intensity at pixel i. A spectral segmentation method with multi-scale graph decomposition is proposed in [4]. It minimizes the $\mathbf{x}^\top A\mathbf{x}/(\mathbf{x}^\top D_g\mathbf{x})$, where A is the affinity matrix and D_g is a diagonal matrix (refer to sec. 3 for descriptions).

Seminal methods [11, 12] derive a matting Laplacian matrix from multiple matte equations. In comparison with random walk and normalized cuts, they adapt a correlation measure instead of the exponent of color distance, a local scaling (inspired by self tuning classification [14]) instead of a global scaling, and formulate a least-squares solution with constraints from user input. Local scaling leads to better clustering especially when the data includes multiple scales and the clusters are placed within a cluttered background. [14] also explores the structure of the eigenvectors to infer automatically the number of groups, instead of checking jumps in eigenvalue magnitudes. [8] proposes a dark-channel prior to model the thickness of the haze and apply the matting Laplacian to refine the transmission map. In the same manner, [9] derives its guided image filtering algorithm from a local linear model similar to the matting Laplacian [11].

3 Laplacian Spectrum Constraint

To estimate the optimal $\mathbf{x}$ from the prior $\mathbf{x}^\star$, we compute a graph Laplacian matrix L from G, where G is a graph representation of the input image $\mathbf{y}$. In other words, the Laplacian matrix L will regularize our under-constrained optimization formulation by laying on the image structure inherent in $\mathbf{y}$. This enables us to define the binary segmentation problem as a least-squares constrained optimization

$$\min_{\mathbf{x}} \|\mathbf{x}-\mathbf{x}^\star\|^2, \quad \text{s.t.} \quad L\mathbf{x}=0. \tag{1}$$

We call the above constraint $L\mathbf{x}=0$ the *Laplacian spectrum* constraint. This is a generalization of the conventional approaches and does not require a specific numerical solver as the matting Laplacian.

Suppose a ‘connected component’ represents a subgraph of G in which any two vertices are connected to each other, yet not connected to any other vertices in the remaining part of the graph. Under image segmentation context, a connected component corresponds regions having the same label. We argue that, the multiplicity of $\lambda=0$ as an eigenvalue of L is equal to the number of connected components in G. This indicates that the spectrum of L determines the number of connected components in G. In other words, it would separate perfectly different connected subgraphs.

To see this, let λ_i be the i-th smallest eigenvalue of L, $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$. Then, we have $\lambda_1=0$ since $L\mathbf{e}=\mathbf{0}$, where $\mathbf{e}$ is the above all-1 vector in $\mathcal{R}^n$. This can be directly derived from the definition of the Laplacian matrix. Suppose the multiplicity of 0 eigenvalue is k, that is, $\lambda_1=\cdots=\lambda_k=0$, and $1 \le k \ll n$. Obviously, k is the dimension of L's null-space null(L) and the k smallest eigenvectors corresponding to these 0 eigenvalues comprise a basis of this null-space. An arbitrary linear transformation of these k eigenvectors would generate another basis.

We are interested in a specific basis such that each of these k orthogonal vectors would have 1 for all the vertices of a component of the graph and 0 for the rest of the vertices, and the sum of these k vectors is $\mathbf{e}$. This ‘ideal’ basis [12, 14, 17], in essence, gives us the perfect segmentation of the input image $\mathbf{y}$. However, due to numerical errors and the limited connectivity of the graph G, one cannot determine k by simply examining the multiplicity of 0 eigenvalue. A more robust way is to search for a significant change in the magnitude of the eigenvalues starting from λ_1. In practice, the numerical stability of estimating k value highly depends on the noise, the data structure, and the construction of G, and thus L.

The graph Laplacian spectrum constraint enforces a given image structure on the prior information (in the data fidelity term $\|\mathbf{x}-\mathbf{x}^\star\|^2$). At the same time, its solver as described in Sec.4 achieves robustness to outliers existing in ordinary applications. With this constraint, the optimal $\mathbf{x}$ should lie in the null-space of L, that is, $\mathbf{x}$ should be constant within each connected component of the graph G. In most cases, the objective binary segmentation results, e.g. foreground and background regions, consist of several disconnected components. Since the estimated $\mathbf{x}$ can be represented by a linear combination of the 0 eigenvectors (or the 'ideal' basis), it is still able to differentiate the foreground components from the background ones. In this way, we can explicitly avoid computing L's nullity k and its basis, while still can use the image structure to regularize the data fidelity term.

4 Alternative Objective Functions

Depending on the norm, several objective functions can be defined. After explaining alternatives, we present a robust formulation.

4.1 Projection onto Null-space

Estimating an optimal $\mathbf{x}$ for the constrained optimization problem Eq.(1) can be considered as a search for a vector in the null-space of L which has the closest distance to the prior $\mathbf{x}^\star$. Let $\mathbf{v}_1,\ldots,\mathbf{v}_k \in \mathcal{R}^n$ be the k eigenvectors of L corresponding to 0 eigenvalue, and let $W = \text{Span}(\mathbf{v}_1,\ldots,\mathbf{v}_k)$ be the k-dimensional subspace of $\mathcal{R}^n$ spanned by these eigenvectors. W is the null-space of L, $\text{null}(L) = W$. Let $V = [\mathbf{v}_1,\ldots,\mathbf{v}_k]$, the optimal solution can be estimated as

$$\mathbf{x} = \text{Proj}_W(\mathbf{x}^\star) = Q\mathbf{x}^\star \ , \tag{2}$$

where Q is the projection matrix for the subspace W, and $Q = V(V^\top V)^{-1}V^\top = VV^\top$.

The assumption here is that the nullity k of L can be determined accurately, which is not always true. Another problem is that this approach approximates $\mathbf{x}$ using $\text{Proj}_W(\mathbf{x}^\star)$, while a solution that is a linear combination of $\mathbf{x}^\star$ and $\text{Proj}_W(\mathbf{x}^\star)$ is more favorable due to noise, limited connectivity of graph G, and computational load. As shown in Fig.2, null-space projection may generates artifacts in the estimated $\mathbf{x}$.

4.2 Convex Function with ℓ_2 Norm on Constraint

Instead of solving a constrained optimization problem Eq.(1), we can transform it into an unconstrained minimization:

$$\min_{\mathbf{x}} \|\mathbf{x}-\mathbf{x}^\star\|^2 + \beta\|L\mathbf{x}\|^2 \ , \tag{3}$$

with a penalty β that enforces the structure in $\mathbf{y}$.

Setting the derivative of the objective function Eq.(3) to 0, we have

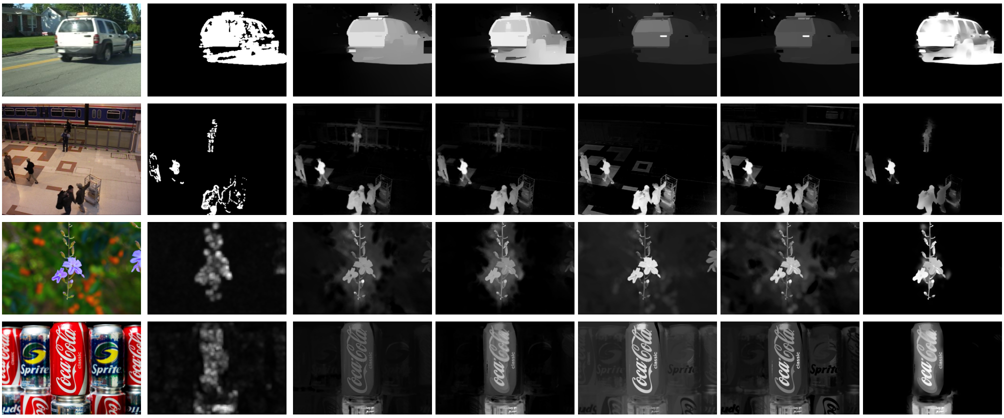
$$\mathbf{x} = (\beta L^\top L + I)^{-1}\mathbf{x}^\star = P\mathbf{x}^\star \ , \tag{4}$$

where I is an identity matrix, and P can be viewed as a modified projection matrix[1].

We draw the connection between P and the previous Q. Since L is a real symmetric matrix, we can diagonalize it as $L = V\Lambda V^\top$, where V is an orthogonal matrix $V = [\mathbf{v}_1,\ldots,\mathbf{v}_n]$ (note that V contains n eigenvectors), and Λ is a diagonal matrix constructed from the eigenvalues of L as $\Lambda = \text{diag}(\lambda_1,\ldots,\lambda_n)$. Therefore, P could be rewritten as:

$$P = V(\beta\Lambda^2 + I)^{-1}V^\top = \sum_{i=1}^{n}\frac{1}{1+\beta\lambda_i^2}\mathbf{v}_i\mathbf{v}_i^\top = Q + \sum_{i=k+1}^{n}\frac{1}{1+\beta\lambda_i^2}\mathbf{v}_i\mathbf{v}_i^\top \tag{5}$$

[1] P does not satisfy the idempotent property of the projection matrix.



input $\mathbf{y}$ weak prior $\mathbf{x}^\star$ $\mathbf{x}$: null(L) $\mathbf{x}$: ℓ_2 $\mathbf{x}$: ℓ_1 $\mathbf{x}$: $D\alpha$ $\mathbf{x}$: robust

Figure 2: Comparison of our proposed solutions. Estimated likelihood weights $\mathbf{x}$ indicating an image pixel belonging to target object using the given prior $\mathbf{x}^\star$. For the top two rows, the prior is binary. For the bottom two rows the prior is a real number in $[0,1]$. In all cases, we use the same Laplacian L computed from $\mathbf{y}$ to solve the introduced optimization $\min_{\mathbf{x}} \|\mathbf{x}- \mathbf{x}^\star\|^2, \text{s.t.} L\mathbf{x}=0$. As visible, the robust function generates the most accurate results since it can remove the negative effects of large outliers when the prior is outside the correct object areas. The ℓ_1 and sparse decomposition ($D\alpha$) have relatively inferior performance since it uses a limited number of eigenvectors.

Eq.(5) indicates that the solution of the penalized least squares problem Eq.(3) is the weighted sum of the projection of $\mathbf{x}^\star$ on each subspace $\mathbf{v}_i\mathbf{v}_i^\top$. Also, P adds influence of non-zero eigenvectors into the final estimate based on their corresponding eigenvalues and penalty term β. If $\beta \rightarrow \infty$, $P=Q$, thus Eq.(3) becomes the constrained least square problem Eq.(1).

4.3 Convex Function with ℓ_1 Norm on Constraint

Instead of enforcing the Laplacian spectrum constraint in the ℓ_2 norm, we can use the ℓ_1 norm to decrease the influence of the large outliers in the noisy prior. In this case, β is not required to approach to ∞ in order to solve the original constrained minimization.

The objective can be rewritten as

$$\min_{\mathbf{x}} \frac{\mu}{2}\|\mathbf{x}-\mathbf{x}^\star\|^2 + \|L\mathbf{x}\|_1 \tag{6}$$

and solved using the Augmented Lagrangian methods, more specifically, alternating direction algorithms [16] in the following iterative framework:

$$\begin{cases} \mathbf{a}^{t+1} \leftarrow \arg\min_{\mathbf{x},\mathbf{a}} \mathcal{L}_{\mathcal{A}}(\mathbf{x}^t,\mathbf{a},\mathbf{c}^t), \\ \mathbf{x}^{t+1} \leftarrow \arg\min_{\mathbf{x},\mathbf{a}} \mathcal{L}_{\mathcal{A}}(\mathbf{x},\mathbf{a}^{t+1},\mathbf{c}^t), \\ \mathbf{c}^{t+1} \leftarrow \mathbf{c}^t - \beta(\mathbf{a}^{t+1} - L\mathbf{x}^{t+1}), \end{cases} \tag{7}$$

where $\mathbf{a}$ is an auxiliary vector, β is a penalty parameter, and $\mathcal{L}_{\mathcal{A}}(\mathbf{x},\mathbf{a},\mathbf{c}^t)$ is the augmented Lagrangian function of Eq.(6) defined as

$$\mathcal{L}_{\mathcal{A}}(\mathbf{x},\mathbf{a},\mathbf{c}) \triangleq \frac{\mu}{2}\|\mathbf{x}-\mathbf{x}^\star\|^2 + \|\mathbf{a}\|_1 - \mathbf{c}^\top(\mathbf{a}-L\mathbf{x}) + \frac{\beta}{2}\|\mathbf{a}-L\mathbf{x}\|^2 , \tag{8}$$

where $\mathbf{c}$ is the Lagrangian parameter vector (slightly abusing the standard Lagrangian notation) that has the same length as $\mathbf{a}$ and $\mathbf{x}$. For each suboptimization, we can solve them directly by

$$\mathbf{a}^{t+1} = \operatorname{sgn}(L\mathbf{x}^t + \frac{1}{\beta}\mathbf{c}^t) \circ \max\left\{\|L\mathbf{x}^t + \frac{1}{\beta}\mathbf{c}^t\| - \frac{1}{\beta}, 0\right\} \tag{9}$$

and

$$\left(L^\top L + \frac{\mu}{\beta} I\right)\mathbf{x}^{t+1} = L^\top(\mathbf{a}^{t+1} - \frac{1}{\beta}\mathbf{c}^t) + \frac{\mu}{\beta}\mathbf{x}^\star, \tag{10}$$

where $\circ$ and sgn represent the point-wise product and the signum function, respectively.

Even though ℓ_1 norm is effective when the error is in the form of the impulsive noise, the regularization error in our case is often continuous and has large values in arbitrary regions of the image (as visible in Fig.2).

4.4 Sparse Decomposition

Another approach to apply the Laplacian spectrum constraint is to analyze its error map, i.e., $\mathbf{x}_{err} = L\mathbf{x}$. An optimal solution of Eq.(1) would have the property that most items of $\mathbf{x}_{err}$ would have 0 value and only a few have large errors, which means we can rewrite Eq.(1) in terms of error sparsity as

$$\min_{\alpha} \|\mathbf{x}^\star - D\alpha\|^2 + \beta\|\alpha\|_1, \tag{11}$$

where $\alpha = L\mathbf{x}$ and the decomposition dictionary D is defined as $D = L^+$ since $\mathbf{x} = L^+\alpha$ and L^+ is the pseudoinverse of L. In this case, we have to compute the explicit inverse of the Laplacian matrix $L^\top L$, which is numerically inaccurate and computationally impractical since L is a large sparse matrix, in order to use the conventional solvers such as Orthogonal Matching Pursuit, LASSO, Elastic-Net, etc. Another problem is that L^+ is no longer a sparse matrix, which requires an extremely large memory space for processing and storage.

Instead of computing L^+ as the decomposition dictionary D, we can construct it directly from the Laplacian spectrum constraint. Recall that, in the ideal case, the optimal $\mathbf{x}$ can be represented by a linear combination of L's 0 eigenvectors, that is, $\mathbf{x} = \sum_{i=1}^{k} \alpha_i \mathbf{v}_i$, where $\mathbf{v}_i$ is the eigenvector corresponding to the i-th smallest eigenvalue $\mathbf{v}_i$ of L.

This property can be easily extended to $\mathbf{x} = D\alpha$ in matrix form, where $D = \left[\mathbf{v}_1, \cdots, \mathbf{v}_{k'}\right]$, $k' \gg k$. As long as k' is much larger than k, we would have a sparse vector α and Eq.(11) can be efficiently solved through the SPAMS toolbox.

In our experiments, we set $k' = 100$ for a better tradeoff between α sparsity and computation efficiency. The equation $\mathbf{x} = D\alpha$ indicates that the final estimate $\mathbf{x}$ is actually an approximated projection of $\mathbf{x}^\star$ on the null-space of L since we limit the number of nonzero values in α and $\mathbf{v}_m$ $(m > k)$ may also contribute to the final estimate $\mathbf{x}$. Compared with the approach, which directly projects $\mathbf{x}^\star$ onto L's null-space, this approach is more robust and can solve the problem Eq.(1) without explicitly computing the nullity k of L.

4.5 Robust Function

Since the residual $\delta = |\mathbf{x} - \mathbf{x}^\star|$ has many spatially continuous large outliers and the least square data fidelity term weights each sample with a quadratic norm, the final estimation of Eq.(3) can be distorted severely. Depending on its quality, the prior information $\mathbf{x}^\star$ could contain incomplete and inaccurate indicators, for instance strong responses across segment boundaries. This may confuse the segmentation algorithm and cause mislabeling.

Algorithm 1: Robust Graph Laplacian Spectrum

```
Input: $L \in \mathcal{R}^{n \times n}$, $\mathbf{x}^\star \in \mathcal{R}^n$, $\beta$
Output: $\mathbf{x} \in \mathcal{R}^n$
$W = I, t = 1$;
while $t < iter_{max}$ & $\|W(\mathbf{x} - \mathbf{x}^\star)\|^2 < err_{max}$ do
    $\mathbf{x}^t = (\beta L^\top L + W)^{-1} W \mathbf{x}^\star$;
    update $W$ according to Eq.(13);
    $t = t + 1$;
return optimal $\mathbf{x}^t$
```

A better option is to weight large outliers less and use the structure information from the Laplacian spectrum constraint to recover the **x**. Therefore, we borrow existing principles from robust statistics [2] and adapt a robust functional to replace the least square cost as

$$\min_{\mathbf{x}} \rho(\mathbf{x} - \mathbf{x}^\star) + \beta \|L\mathbf{x}\|^2 , \tag{12}$$

where ρ is a robust function that could be the Huber function [10], Cauchy, ℓ_1, or other M-estimators. We choose to use the Huber function since it is a parabola in the vicinity of 0 and increases linearly when δ is large. Thus, the effects of large outliers can be eliminated significantly. Here, we define the weight function $\mathbf{w} = [w_1, \ldots, w_n]^\top$ at each pixel associated with the Huber function as

$$w_i = \begin{cases} 1 & \text{if } \delta_i < \varepsilon \\ \varepsilon / \delta_i & \text{if } \delta_i \geq \varepsilon \end{cases} . \tag{13}$$

When written in matrix form, we use a diagonal weighting matrix $W = \mathrm{diag}(w_1, \ldots, w_n)$ to represent the Huber weight function. Therefore, the data fidelity term can be simplified as $\rho(\mathbf{x} - \mathbf{x}^\star) = \|W(\mathbf{x} - \mathbf{x}^\star)\|^2$. As a result, the problem Eq.(12) can be solved efficiently in an iterative least square approach. At each iteration, the optimal **x** is updated as

$$\mathbf{x} = (\beta L^\top L + W)^{-1} W \mathbf{x}^\star . \tag{14}$$

The details of our algorithm are shown in Alg.1. To initialize the algorithm, we could set $W = I$ for the first iteration, or when a confidence measure $m^\star$ is available for $\mathbf{x}^\star$, we could directly use $m^\star$ to initialize W. Note that, if we fix W to be I, then Eq.(14) is exactly the same as Eq.(4).

5 Seeded Segmentation & Larger Windows

Various forms of the graph Laplacian matrix have been adopted for different applications such as image segmentation by normalized cuts [15], image segmentation by random walks [7], data classification [14], and matte estimation [11]. In Eq.(13), if we set $w_i = 0$ for $\delta_i \geq \varepsilon$, then the solution Eq.(14) becomes similar to the close-form matting [11] and the random walks segmentation [7]. Defining the term $W\mathbf{x}^\star$ in Eq.(14) as an updated observation $\mathbf{x}^\star_{new} \leftarrow W\mathbf{x}^\star$ and setting $w_i = 0$ means that the updated prior $x^\star_{i,new}$ corresponding to $w_i = 0$ does not contribute to the least square estimation of **x** (note that $\mathbf{x}^\star_{new} = [x^\star_1, \ldots, x^\star_n]_{new}$). Thus,

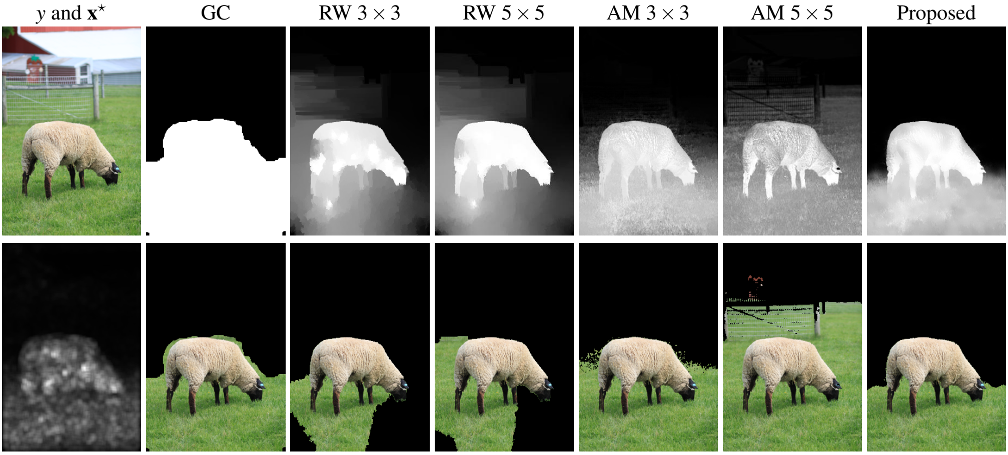


Figure 3: Infocus segmentation results of graph-cuts (GC), random walk (RW), alpha matting (AM), and Laplacian spectrum constraint with robust function. $\mathbf{x}^\star$ is our input in-focus confidence. Note that doubling the window size for RW and AM does not necessarily generate better results. Top row shows the confidence maps (likelihood of being in-focus). Bottom row shows the segmented focused regions. Our map is more consistent with the underlying image structure and visibly superior to both same and double-window size versions of the random walk and alpha matting methods.

the algorithm recovers x_i based on the local structure information through L. The assignment $w_i = 1$ indicates these updated priors $x^\star_{i,new}$ will serve as seed points. This will propagate the corresponding values into the unknown regions according to the Laplacian matrix.

The difference between the presented robust approach and the aforementioned seeded segmentation methods is the use of $L^\top L$ instead of L. Firstly, $L^\top L$ is **no** longer a Laplacian matrix at all. Under the orthogonal projection context, $L^\top L$ shares the same eigenvectors as L except that its eigenvalues are the squared ones of L. As discussed in the Sec.4.2 Eq.(5), our approach favors the projection bases associated with small eigenvalues given the same penalty parameter β. In other words, the large eigenvectors in our algorithm have less influence on the final estimation results. Therefore, our algorithm tends to generate more accurate results since the small eigenvectors are of more importance to enforce the structure onto the final result $\mathbf{x}$.

Note that, doubling support window for L generate different constraints from the robust functional norm $L^\top L$. Thus, using a larger window with the $\mathbf{x}^\top L\mathbf{x}$ cost will not produce same effect of the $\mathbf{x}^\top L^\top L\mathbf{x}$ cost. This can also be seen in Fig. 3. Suppose we are given a simple 1D chain graph $x_1 \leftrightarrow x_2 \leftrightarrow x_3 \leftrightarrow x_4$. The $x^\top L\mathbf{x}$ cost is $(x_1-x_2)^2+(x_2-x_3)^2+(x_3-x_4)^2$. The cost implied by doubling the support window for $\mathbf{x}^\top L\mathbf{x}$ is the sum the first order derivatives: $(x_1-x_2)^2+(x_1-x_3)^2+(x_2-x_3)^2+(x_2-x_4)^2+(x_3-x_4)^2$. However, for $\mathbf{x}^\top L^\top L\mathbf{x}$, we have $(x_1-x_2)^2+(-x_1+2x_2-x_3)^2+(-x_2+2x_3-x_4)^2+(x_3-x_4)^2$, which imposes both first (on the boundary) and second order derivatives in addition to having a doubling support window effect. The fundamental difference between the robust functional norm $L^\top L$ and Laplacian matrix with doubling support window lies at the different order of derivatives. In addition, $L^\top L$ assumes linear smoothness, that is, $x_i = (x_{i-1}+x_{i+1})/2$.

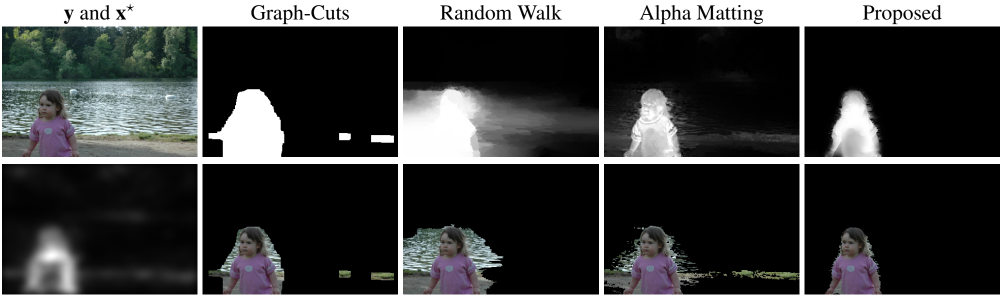


Figure 4: Saliency segmentation results of graph-cuts, random walk, alpha matting, and Laplacian spectrum constraint wit robust function. Top row shows the confidence maps (likelihood of being more salient). Bottom row shows the segmented regions.

	GC [3]	RW [7]	AM [11]	Robust $L\mathbf{x}$
Sheep:	0.94	0.81	0.93	**0.97**
Girl:	0.81	0.63	0.79	**0.95**
Boat:	0.89	0.71	0.85	**0.95**
Farm:	0.63	0.61	0.89	**0.96**
Flower:	0.73	0.80	**0.88**	0.87
Depth:	0.84	0.56	0.95	**0.99**

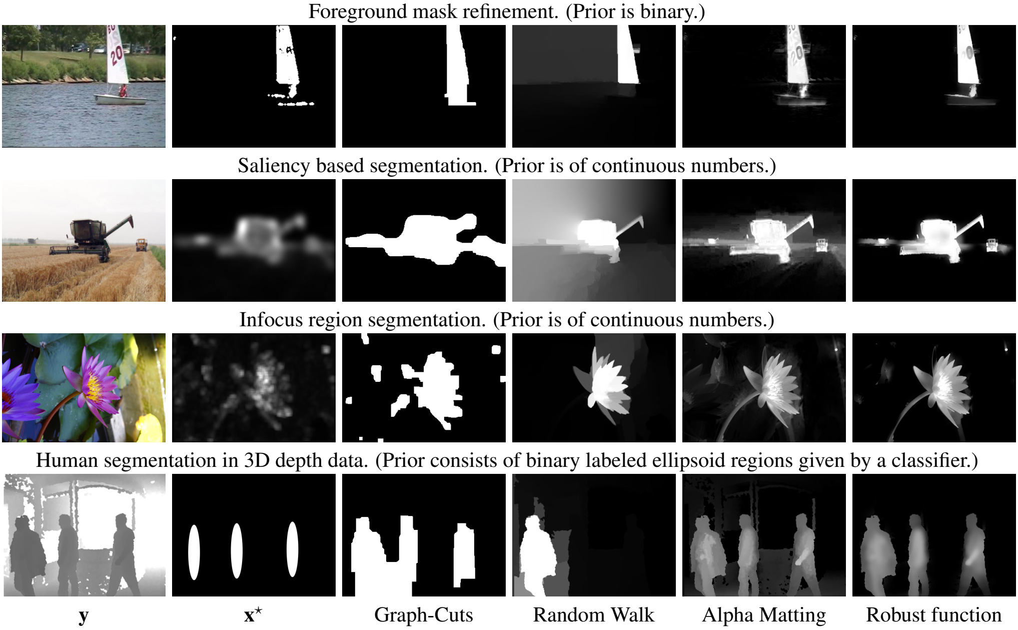
Table 1: F-measure scores. GC: graph-cuts, RW: random walk, AM: alpha matting.

6 Experimental Analysis

The Laplacian spectrum constraint $L\mathbf{x} = 0$ is applicable to any segmentation problem that comes with a (weak or rigid) prior. To illustrate possible usages, we apply our novel objective functions to 1) refine noisy and incomplete foreground masks in change detection [1], 2) segment object boundaries using noisy saliency results [6], 3) segment infocus regions[13], and 4) segment objects detected by a classifier.

For the most objective evaluations, we compare our segmentation results with the best results of graph-cuts [3], random walk [7], and alpha matting [11] after a fine-tuning of each of these algorithms. As visible in Fig.3 and 4, graph-cuts fails along the sheep body since the defocus prior $\mathbf{x}^\star$ has a strong response across the object boundaries. Note that, graph-cuts generates a binary confidence. Random walk confidence map is not correct as it cut through the grass failing to impose the image structure, in this case color similarity. Alpha matting confidence map has noise and inaccurate granular details, e.g. in the mid-field it contains parallel texture. Note that increasing from window size from 3×3 to 5×5 for other methods does not necessarily generate better results, or results similar to our method.

When we compare our different solvers, performances from the less accurate to the best are sparse, null-space, ℓ_1, ℓ_2, and robust function. Sparse decomposition (both LASSO and OMP versions) has unstable performance. This is mainly due to the use of limited number of eigenvectors in the dictionary $\mathbf{x} = D\alpha$. As a result, it is almost impossible to get clean backgrounds. Null-space projection formulation also uses a linear combination of eigenvectors to represent the final solution, yet it uses more eigenvectors. We determined the k by find significant jump in the eigenvalue values starting from the smallest one. This value is not very critical as eigenvectors associated with eigenvalues larger than λ_k do not



Foreground mask refinement. (Prior is binary.)

Saliency based segmentation. (Prior is of continuous numbers.)

Infocus region segmentation. (Prior is of continuous numbers.)

Human segmentation in 3D depth data. (Prior consists of binary labeled ellipsoid regions given by a classifier.)

Figure 5: Several applications of imposing point-wise constraints to a given prior. For each example, we only show the confidence maps (labels). Segmentation results are obtained by simply thresholding. The threshold is fixed to 0.1 for all our results, while other methods require fine-tuned thresholds for each individual image. Another problem about the estimated alpha matting confidence maps is that they are noisy along object boundaries (and sometimes inside the object region).

contribute to the final solution that much. The projection of $\mathbf{x}^\star$ on each eigenvector space $\mathbf{v}_i^\top \mathbf{v}_i (i > k)$ could have small effects. We observed that $k = 50$ is a reasonable choice. As we discussed before, ℓ_2 favors the affine combination of the projection on V and the original $\mathbf{x}^\star$, which could generate a smaller data fidelity term error by slightly violating the Laplacian spectrum constraint because it deviates from the projection space. The motivation of using robust function in the data fidelity term is to remove the effects of outliers.

Table 1 presents the F-measure scores of segmentation results for graph-cuts, random walk, alpha matting, and our robust function formulation on several images. F-measure is defined as $2\frac{Pr \cdot Re}{Pr + Re}$ where Pr is the precision $TP/(TP+FP)$ and Re is the recall $TP/(TP+FN)$ ratios. TP, FP, FN are true positives, false negatives, and false negatives. As shown, our proposed method consistently gives high scores, and most cases the best results.

7 Conclusion

We have presented a novel formulation to impose image structure on a given prior map. We showed that a robust formulation of this constraint generates superior results in binary segmentation. We will extend our work to multi-label segmentation as a future direction.

References

[1] www.changedetection.net.

[2] M. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1):57–91, 1996.

[3] Y. Boykov and G. Funka-Lea. Graph cuts and efficient ND image segmentation. *International Journal on Computer Vision*, 70(2):109–131, 2006.

[4] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. *CVPR*, 2005.

[5] D. Cvetkovic, P. Rowlinson, and S. Simic. *An introduction to the theory of graph spectra*. Cambridge University Press, 2010.

[6] E. Erdem and A. Erdem. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of Vision*, 13(4):1–20, 2013.

[7] L. Grady. Random walks for image segmentation. *PAMI*, 28(11):1768–1783, 2006.

[8] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *CVPR*, 2009.

[9] K. He, J. Sun, and X. Tang. Guided image filtering. *ECCV*, 2010.

[10] P. Huber. *Robust statistics. 1981*. Wiley, New York.

[11] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *PAMI*, 30(2):228–242, 2008.

[12] Anat Levin, Alex Rav Acha, and Dani Lischinski. Spectral matting. *PAMI*, 30(10): 1699–1712, 2008.

[13] F. Li and F. Porikli. Harmonic variance: A novel measure for in-focus segmentation. *BMVC*, 2013.

[14] P Perona and L Zelnik-Manor. Self-tuning spectral clustering. *NIPS*, 17:1601–1608, 2004.

[15] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[16] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM Journal on Scientific Computing*, 33(1):250–278, 2011.

[17] S. Yu and J. Shi. Multiclass spectral clustering. *ICCV*, 2003.