SEEDED LAPLACIAN: AN INTERACTIVE IMAGE SEGMENTATION APPROACH USING EIGENFUNCTIONS

Ahmed Taha and Marwan Torki

Engineering Mathematics - Computer and Systems Engineering
{ahmed.taha,mtorki}@alexu.edu.eg

ABSTRACT

In this paper, we cast the scribbled-based interactive image segmentation as a semi-supervised learning problem. Our novel approach alleviates the need to solve an expensive generalized eigenvector problem by approximating the eigenvectors using a more efficiently computed eigenfunctions. The smoothness operator defined on feature densities at the limit $n \to \infty$ recovers the exact eigenvectors of the graph Laplacian, where n is the number of nodes in the graph.

In our experiments scribble annotation is applied, where users label few pixels as foreground and background to guide the foreground/background segmentation. Experiments are carried out on standard data-sets which contain a wide variety of natural images. We achieve better qualitative and quantitative results compared to state-of-the-art algorithms.

***Index Terms*—** interactive segmentation, eigenfunctions, vision, graph Laplacian

1. INTRODUCTION

Image segmentation is an important problem in computer vision, and it is usually an intermediate step in image processing. Image segmentation divides an image into a small set of meaningful segments, which simplify any further analysis. Due to the variety of image segmentation applications and the specialized requirements for each application, segmentation literature witnesses advances on how to approach such problem. Some of these approaches are normalized-cut [1], mean-shift [2], Graph-cut [3], etc. Each method has pros and cons in terms of the result's quality, processing time, and user-specified parameters. Due to the difficulty of fully automatic image segmentation, user-interactive segmentation is usually introduced to relax the segmentation problem for certain applications.

In interactive image segmentation, users guide the segmentation process by providing annotations. User-specific annotations can take various forms, like bounding box, sloppy contour, and scribbles. Although, scribbles are often favored due to their ease of use in terms of time and effort, scribbles generally provide less information than that provided by bounding box or sloppy contour. There is always a compromise between the choice of annotation type in terms of speed and time and its effect on the segmentation process in terms of quality and accuracy. A recent study [4] presents common annotation forms comparing the tradeoffs and advantages of each form.

Scribble segmentation methods can be categorized into two main categories: **region growing-based** methods and **graph-based** methods. In region growing methods, an iterative approach is employed to label unlabeled pixels near the labeled ones. This iterative process ends when all pixels are labeled as either foreground or background pixels. Known examples for the region growing methods includes Maximal Similarity based Region Merging (MSRM) [5] and Seeded region growing [6]. However, region growing methods encounter some drawbacks. For example, they don't have a clear cost function. They also suffer when the foreground or background regions are not connected and require extra user annotation to overcome such limitation. Being iterative is yet another computational limitation for these methods, but using super-pixels is a typical workaround for these limitations.

On the other hand, graph-based methods like Normalized Cuts [1] and Boykov Jolly [3] have clear cost function; they don't suffer the unconnected regions problem, but they are computationally expensive. Fortunately, fast implementations of polynomial graph cut algorithms are available, like max-flow [7], push-relabel [8], and eigenvector approximation for graph Laplacian [1].

In this paper, we introduce a *novel* scribble-based interactive image segmentation algorithm. In our algorithm we propose to calculate eigenfunctions to approximate Laplacian eigenvectors. Such trick reduces the space and time, needed to build and solve the graph-based labeling process considerably, from minutes to seconds. To the best of our knowledge, we are the first to solve the scribble-based segmentation problem using efficiently computed eigenfunctions.

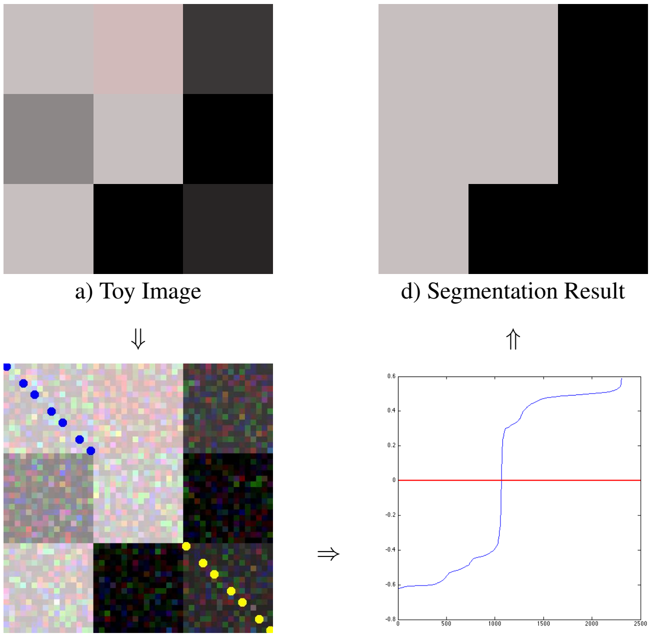
Our formulation brings three ***key contributions*** for the problem:
1) Scalability: The exact eigenvectors computation of graph Laplacian is space and time consuming. So we propose to compute the eigenfunctions and interpolate them to obtain the eigenvectors. This drastically reduces the time needed to a real time performance.
2) Accuracy: Seeded Laplacian (SL) achieves highly competitive results against state-of-the-art interactive image segmentation methods. It can be submerged with more sophisticated algorithms as well.
3) Flexibility to feature type: Seeded Laplacian supports different pixel features like spatial information, different color spaces, depth, texture, and spatial gradients.

In Figure 1, we overview our approach. We show that even with moderate Gaussian noise, the eigenfunction calculation is robust enough to produce the correct segmentation.

The rest of the paper is structured as follows. Sections 2 and 3 present the Seeded Laplacian approach in details. In section 4 we demonstrate our experiments and compare Seeded Laplacian results with other segmentation frameworks. Finally, in Section 5 we present our conclusions and potential future work.

2. SEMI-SUPERVISED LEARNING

In our approach, we cast the interactive image segmentation problem as a semi-supervised learning problem. Following the notations of Zhu et al [9], the user provides labeled points (pixels in our case) of input-output pairs $(X_l, Y_l) = \{(x_1, y_1), ..., (x_l, y_l)\}$ and unlabeled pixels $X_u = \{x_{l+1}, ..., x_n\}$. In our problem, $Y_l \in \{B, F\}$, where B denotes a background label while F denotes a foreground label.



a) Toy Image

d) Segmentation Result

b) Perturbed Image with the user annotations

c) Zero threshold eigenfunction

Fig. 1: Seeded Laplacian approach overview on Toy Image of size 48 × 48. (b)Toy image is perturbed with Gaussian noise and overlaid by seeded annotations (blue for background and yellow for foreground). (c)Setting a Zero threshold on the eigenfunction vector produces (d) the final segmentation result. Best seen in color.

A very common approach in semi-supervised learning is to use a graph-based algorithm. In graph-based methods , a graph $G = (V, E)$ is constructed where the vertices V are the pixels $x_1, ..., x_n$, and the edges E are represented by an $n \times n$ matrix W . Entry W_{ij} is the edge weight between pixels x_i, x_j and a common practice is to set $W_{ij} = exp(-\|x_i - x_j\|^2/2\epsilon^2)$. Let D be a diagonal matrix whose diagonal elements are given by $D_{ii} = \sum_j W_{ij}$, the combinatorial graph Laplacian is defined as $L = D - W$, which is also called the un-normalized Laplacian. A common objective function will have the following form:

$$J(f) = f^T L f + \sum_{i=1}^{l} \lambda (f(i) - y_i)^2 \tag{1}$$

$$= f^T L f + (f - y)^T \Lambda (f - y) \tag{2}$$

The first term in eq. 1 controls the smoothness of the labeling process. This would ensure the estimated labels $f_i's$ will not change too much for nearby features in the feature space. The second term penalizes the disagreement between the estimated labels $f_i's$ and the original labels $y_i's$ that are given to the algorithm.

Λ is a diagonal matrix whose diagonal elements $\Lambda_{ii} = \lambda$ if i is a labeled pixel and $\Lambda_{ii} = 0$ for unlabeled pixels. The minimizer of eq. 1 is the solution of $(L + \Lambda)f = \Lambda y$. To reduce the complexity of the problem, we choose to work with a small number of eigenvectors with the smallest eigenvalues as suggested by [10, 9, 11].

As noted by [14], eq. 1 can be further simplified to the following form.

$$\alpha = \left(\Sigma + U^T \Lambda U\right)^{-1} (U^T \Lambda y) \tag{3}$$

where α are the eigenvalues, U is a matrix of the eigenvectors interpolated from eigenfunctions computed by eq. 4, and $\Sigma = U^T L U$.

Different from the implementation of [14], our implementation approach is to reduce the computational cost of ΛU multiplication operation by a simple scalar multiplications. We define $U_{labeled}$ as a sub-matrix of U containing the rows corresponding to the labeled pixels. Then, $U_{labeled}$ is multiplied by λ scalar value as $\lambda U_{labeled}$. A new Zero matrix of $size(\Lambda U)$ is constructed and the result of $\lambda U_{labeled}$ is inserted into the Zero matrix. Similar approach is applied to reduce the $\mathbf{\Lambda y}$ computational cost.

2.1. Eigenfunction Approach

Like [12, 13, 14], we assume $x_i's \in \Re^d$ are samples from a distribution $p(x)$. This density defines a weighted smoothness operator on any function $F(x)$ defined on $\Re^d$ which we denote by $L_p(F) = \frac{1}{2}\int (F(x_1) - F(x_2))^2 W(x_1, x_2) p(x_1) p(x_2) dx_1 x_2$ Where, $W(x_1, x_2) = \exp(\|x_1 - x_2\|^2 / 2\epsilon^2)$.

According to [14], under suitable convergence conditions the eigenfunctions of the smoothness operator $L_p(F)$ can be seen as the limit of the eigenvectors for the graph Laplacian L as the number of points goes to infinity.

The eigenfunction calculation can be solved analytically for certain distributions. A numerical solution can be obtained by discretizing the density. Let g be the eigenfunction values at a set of discrete points, then g satisfies:

$$(\tilde{D} - P\tilde{W}P)g = \sigma P \hat{D} g \tag{4}$$

where σ is the eigenvalue corresponding the the eigenfunction g, $\tilde{W}$ is the affinity between the discrete points, P is a diagonal matrix whose diagonal elements give the density at the discrete points, and $\tilde{D}$ is a diagonal matrix whose diagonal elements are the sum of the columns of $P\tilde{W}P$, $\hat{D}$ is a diagonal matrix whose diagonal elements are the sum of the columns of $P\tilde{W}$. For every eigenfunction calculated, an 1D interpolation will be done at the labeled points x_l.

The solution for eq 4 will be a generalized eigenvector problem of size $b \times b$, where b is the number of discrete points of the density. Since $b << n$,there is no need to construct the graph laplacian matrix L or to solve for a more expensive generalized eigenvector problem.

In image segmentation problem, the size of image can grow high, while the complexity of solving $b \times b$ system stays unaffected. Another interesting point is that the growth of image size will not affect the quality of the eigenfunction. On the other hand as $n \to \infty$ the eigenfunction will approximate better the generalized eigenvector solution of L.

As noted by [14], eigenfunction solution will exactly recover the eigenvector solution of the graph laplacian L if the feature dimensions are independent. To have independent components we rotate the data using principal component analysis (PCA) [15].

3. APPROACH

In this section we provide details for the features used in our approach as well as the algorithmic details of the Seeded Laplacian algorithm.

3.1. Feature Types

1)Color Features: It is a common practice to use the color of the pixels to represent the appearance. We use **RGB** and **Lab** color spaces to represent every pixel by a six elements vector.

2)Spatial Features: Neighboring pixels should not change their label abruptly. Thus, penalizing the label change by the spatial distance is important. We use a Gaussian decaying function for that purpose.
3)Contour Features: We use the intervening contour idea of [16]. Intervening contour usage appeared to be very useful in interactive image segmentation. It provides a vehicle to separate neighboring pixels having similar color intensities if a boundary exists between them.

Spatial features and contour features are computed around certain points. Thus, to augment the feature types mentioned earlier, we sample uniformly the foreground and background scribbles for a representative set of pivots. We use k_1 and k_2 samples from foreground and background scribbles, respectively. We compute feature vectors by measuring the pixel-pivot affinity. We base pixel-pivot affinity on a number of pixels' features: (1) RGB; (2)Lab color; (3) spatial proximity; and (4) intervening contour.

Feature augmentation: We employed two ways to augment the pixel-pivot affinities.

Feature concatenation. For every pixel, we compute a color affinity to the pivots, and do the same for the spatial proximity and the intervening contour affinity. This will end up with a vector of size $3 * (k_1 + k_2)$ for every pixel.

Feature multiplication. Instead of concatenating the color, spatial, and intervening contour affinities, we multiply them together. The multiplication result affinities are concatenated with the original color features for the pixel. This will result in a compact vector of size $k_1 + k_2 + 6$ for every pixel. [17] has shown that product kernels tend to produce better results for kernel combination in recognition problems. Both ways are used in the experimental evaluation, with better results for the multiplication affinity approach.

3.2. Algorithm

We summarize the algorithm in algorithm 1.

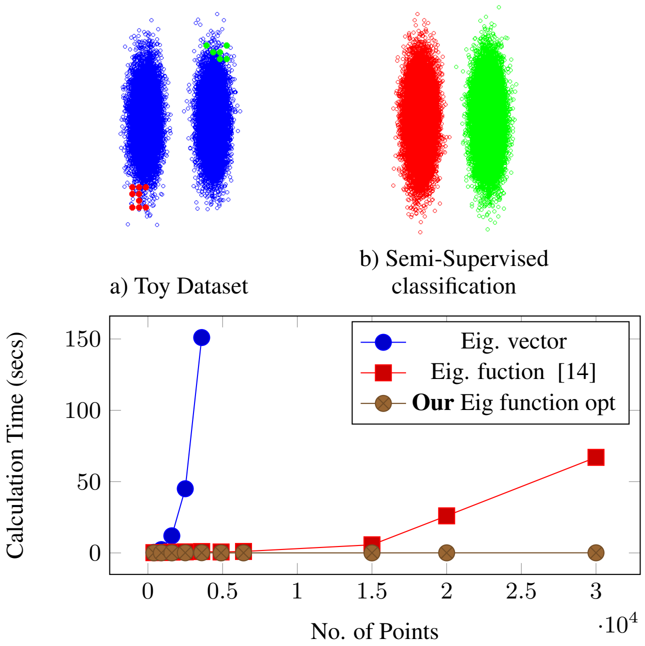
Algorithm 1: Algorithm for Seeded Laplacian (SL)

```
Data: $(X_l, Y_l)$for scribbled annotation, $k_1$and$k_2$ are numbers
      of the fg and bg pivots, $m$ the number of
      eigenfunctions
Result: Binary classification for all pixels in the image

1-Feature extraction
    a- Sample $k_1$ fg pivots and $k_2$ bg pivots.
    b- Compute pixel to pivot affinities (color, spatial and
    intervening contour).
    c- Augment RGB and Lab pixel features with pixel to pivot
    affinities using concatenation or product.

2-Eigenfunction Calculation
    a- Rotate the feature vectors using PCA.
    b- Sample the features densities at $b$ bins.
    c- Compute the first smallest $m$ eigenfunctions. Eq. 4
    d- Interpolate around labeled points$(X_l)$.

3-Compute segmentation
    a- Repeat steps 1,2 for different proximity scales
    b-Set Threshold at Zero to find $+ve/-ve$ segments.
    c-Apply post processing to remove small islands like [22].
    d- Output the binary labels for the pixels.
```


a) Toy Dataset

b) Semi-Supervised classification

Fig. 2: Time Complexity Analysis.

4. EXPERIMENTAL RESULTS

4.1. Time Complexity Analysis

We present a brief time complexity analysis in Figure 2. We sample points from two Gaussian distributions and solve for the semi-supervised classification. We vary the number of samples and check the time needed to compute the eigenfunction solution of eq. 4 versus the eigenvector solution of eq. 1. We implemented our optimized code for the eigenfunction solution and we called it eigenfunction optimized.

4.2. Comparative Evaluation

Dataset: The Geodesic Star-Dataset [18] is a recent scribble-based interactive image segmentation dataset. The dataset consists of 151 images: 49 images taken from the GrabCut dataset [19], 99 from the PASCAL VOC dataset [20], and 3 images from the Alpha matting dataset [21]. For every image, the dataset simulates a user input in the form of an image containing four user scribbles. These four fixed scribbles are divided into one foreground and three background scribbles.

We extended the dataset in order to simulate a user input of one or two background scribbles only, to measure performance degradation in terms of accuracy with various background scribble inputs, as shown Figure3.
Evaluation Measures: In our experiments, the Jaccard Index (a.k.a. overlap score) and F-score are used to measure the accuracy of different segmentation algorithms. Jaccard Index is a measure used to evaluate segmentation quality in the VOC segmentation challenge [20].
Quantitative Evaluation: We quantitatively compare our proposed method with multiple algorithms, including BJ, PP, SP-IG, SP-LIG, and SP-SIG [3] [22] [18] which gives best performance on scribble segmentation reported by [18]. GSCseq and ESC are demonstrated as the state-of-the-art [18]. In all experiments, we set number of

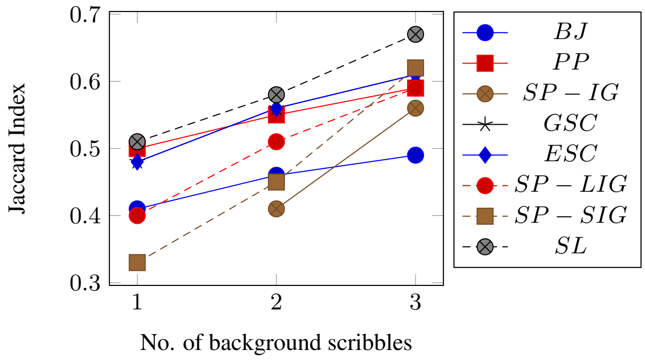


Fig. 3: Comparing interactive segmentation methods' performance against number of background scribbles provided by the user scribble annotation. SL achieved better Jaccard Index against competitive methods for every count of background scribbles.

eigenfunctions to 100, number of foreground pivots to 21, and number of background pivots to 21.

Method Name	Jaccard Index	F Score
BJ	0.49 ± 0.26	0.62
PP	0.59 ± 0.25	0.71
GSC	0.61 ± 0.25	0.72
ESC	0.61 ± 0.24	0.72
SP-IG	0.56 ± 0.16	0.70
SP-LIG	0.59 ± 0.22	0.72
SP-SIG	0.62 ± 0.17	0.75
SL Concatenation	0.65 ± 0.17	0.77
SL Multiplication	**0.67** ± 0.18	**0.79**

Table 1: Comparative Evaluation Experiment. The performance of various scribbled segmentation algorithms are compared using same Geodesic Star-Dataset input annotations. In such experiment, each algorithm is presented by an annotation image which contain one label as foreground and three other labels as background.

The evaluation measures in terms of Jaccard Index and F-score is presented in Table 1. Our quantitative experiment confirms multiple factors.

1)The stability of our approach, where we see that the standard deviation in the Jaccard index is 18%. That is very close to both SP-SIG (17%) and SP-IG (16%). Other methods, like GSC and ESC, presented as state-of-the-art in [18], suffer standard deviation higher than 24%.

2) The robustness against fewer scribbles is illustrated in Figure 3. It is remarkable that SP-SIG performance decreased to 33% when the user provided annotation has only one background label as shown in Figure 3. While SP-IG was not able to calculate a solution at all[1].

3) Our algorithm produces accurate segmentation; our F-score is better when compared to other counterparts.

4) The feature types presented in section 3 are well-used by our algorithm. Our results show the multiplication affinity approach produced better results. We also note that concatenated features comes second in Table 1.

Qualitative Evaluation: In Figure 4 we show our results against the SP-SIG which comes next to SL in the comparative evaluation experiment.

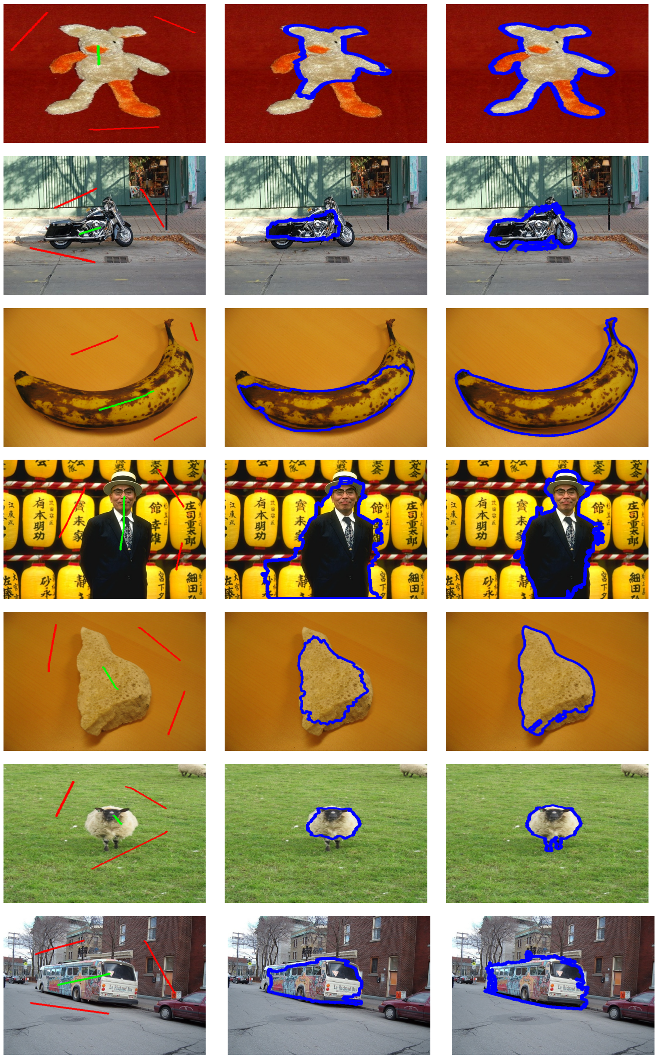


Fig. 4: Qualitative Results for 7 out of 151 images. The first column shows the original image with user scribble annotation. The second column shows the segmentation results using SP-SIG method. The third column shows the segmentation results using SL method

5. CONCLUSION AND FUTURE WORK

We presented a novel scribble-based interactive image segmentation approach. Interactive image segmentation problem was cast as a graph-based semi-supervised learning problem. We relaxed the need to work on the exact graph by means of eigenfunctions. The results in the previous section provide quantitative and qualitative evidence for the competitiveness of the algorithm against state-of-the-art algorithms. Spatial features and intervening contours integration appeared to be a worthy addition in our experiments. [2]

[1] We use the original codes by the authors of [18]

[2] The MATLAB source code and the paper's supplementary material will be available at `ahmed-taha.com` and `eng.alexu.edu.eg/~mtorki/`.

6. REFERENCES

[1] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[2] Dorin Comaniciu and Peter Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[3] Yuri Y Boykov and M-P Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Eighth IEEE International Conference on Computer Vision, ICCV*, 2001.

[4] Suyog Dutt Jain and Kristen Grauman, "Predicting sufficient annotation strength for interactive foreground segmentation," in *IEEE International Conference on Computer Vision (ICCV).*, 2013.

[5] Jifeng Ning, Lei Zhang, David Zhang, and Chengke Wu, "Interactive image segmentation by maximal similarity based region merging," *Pattern Recognition*, 2010.

[6] Rolf Adams and Leanne Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994.

[7] LR Ford and Delbert Ray Fulkerson, *Flows in networks*, Princeton Princeton University Press, 1962.

[8] Andrew V Goldberg and Robert E Tarjan, "A new approach to the maximum-flow problem," *Journal of the ACM (JACM)*, 1988.

[9] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al., "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003.

[10] Bernhard Schölkopf and Alexander J Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2002.

[11] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al., *Semi-supervised learning*, MIT press Cambridge, 2006.

[12] Boaz Nadler, Stéphane Lafon, Ronald R Coifman, and Ioannis G Kevrekidis, "Diffusion maps, spectral clustering and reaction coordinates of dynamical systems," *Applied and Computational Harmonic Analysis*, 2006.

[13] Yair Weiss, Antonio Torralba, and Rob Fergus, "Spectral hashing," in *Advances in neural information processing systems*, 2009.

[14] Rob Fergus, Yair Weiss, and Antonio Torralba, "Semi-supervised learning in gigantic image collections," in *Advances in neural information processing systems*, 2009.

[15] Ian Jolliffe, *Principal component analysis*, Wiley Online Library, 2005.

[16] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi, "Contour and texture analysis for image segmentation," *International journal of computer vision*, 2001.

[17] Manik Varma and Bodla Rakesh Babu, "More generality in efficient multiple kernel learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009.

[18] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman, "Geodesic star convexity for interactive image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[19] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM Transactions on Graphics (TOG)*, 2004.

[20] M Everingham, L Van Gool, C Williams, J Winn, and A Zisserman, "The pascal visual object classes challenge 2009," in *2th PASCAL Challenge Workshop*, 2009.

[21] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott, "A perceptually motivated online benchmark for image matting," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2009.

[22] Jiangyu Liu, Jian Sun, and Heung-Yeung Shum, "Paint selection," in *ACM Transactions on Graphics (ToG)*, 2009.