

# Learning Graph Laplacian for Image Segmentation

Sergey Milyaev  
Radiophysics Department  
Voronezh State University, Voronezh, Russia  
sergey.milyaev@gmail.com

Olga Barinova  
Department of Computational Mathematics and Cybernetics  
Lomonosov Moscow State University, Moscow, Russia  
obarinova@graphics.cs.msu.ru

## Abstract

In this paper we formulate the task of semantic image segmentation as a manifold embedding problem and solve it using graph Laplacian approximation. This allows for unsupervised learning of graph Laplacian parameters individually for each image without using any prior information. We perform experiments on GrabCut, Graz and Pascal datasets. At a low computational cost proposed learning method shows comparable performance to choosing the parameters on the test set. Our framework for semantic image segmentation shows better performance than the standard discrete CRF with graph-cut inference.

**Keywords:** *semantic image segmentation, unsupervised learning, manifold embedding.*

## 1. INTRODUCTION

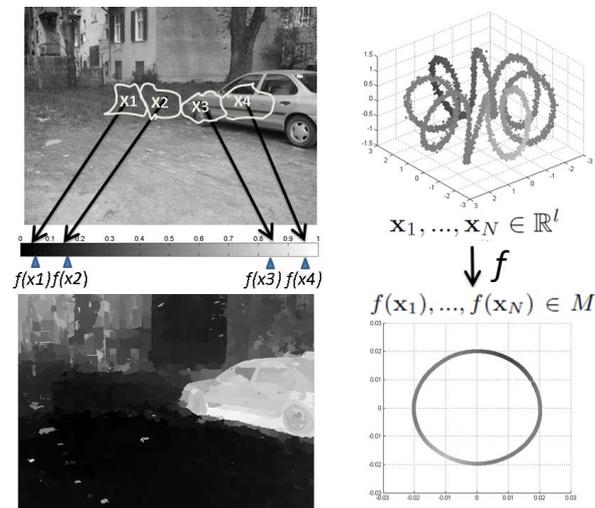
We consider the task of semantic image segmentation that implies assigning a label from a given set to each image pixel. Various discrete CRF models have been proposed for this task [1] [2], [3]. It was shown that learning the parameters of CRF improves its performance [4], [5]. In this work we propose an alternative view on semantic image segmentation.

Methods based on graph Laplacians show state-of-the-art results for interactive image segmentation [6] and image matting [7]. They require just a few local computations and solving one sparse linear system, which can be done very efficiently. In this work we propose a formulation of image segmentation task in terms of manifold embedding and discretize the problem using graph Laplacian approximation.

Graph Laplacian methods have the parameters very similar to those of discrete CRFs. While a remarkable progress has been done in the direction of learning the parameters of discrete CRFs (see e.g. [4], [5]). However the methods for learning parameters of discrete CRF are not applicable to graph Laplacian. Thus the parameters of graph Laplacian are usually chosen by validation on hold-out dataset. The use of validation limits the potential number of parameters used. Moreover, the optimal values of parameters can vary significantly from one image to another, therefore choosing the parameters individually for each image is desirable.

Our formulation of image segmentation problem leads to a novel method for unsupervised learning of graph Laplacian parameters, which is the main contribution of this paper. Our method is designed specifically for the task of semantic image segmentation and provides the values of parameters individually for each test image without using any kind of supervision. Proposed method is computationally efficient and achieves performance comparable to choosing the parameters on the test set, which eliminates the need of using hold-out set or cross-validation. In experimental comparison on Graz and Pascal datasets proposed method shows better performance than the standard discrete CRF with graph-cut inference.

The remainder of the paper is organized as follows. We start by discussing related work. In section 3 we describe the image segmentation framework proposed in this paper. In section 4 we present our method for unsupervised learning of graph Laplacian parameters. We proceed to the experimental evaluation of the



**Figure 1:** We formulate semantic image segmentation task as one-dimensional manifold embedding problem. This allows for unsupervised learning of graph Laplacian parameters individually for each image.

proposed method.

## 2. RELATED WORK

The task of semantic image segmentation implies assigning a label from a given set to each image pixel. Various discrete CRF models have been proposed for this task [1] [2], [3]. Learning the parameters of CRF can improve performance of semantic image segmentation [4], [5]. In this work we use an alternative formulation of semantic image segmentation problem that leads to using graph Laplacian instead of discrete CRF.

Methods based on graph Laplacians have emerged recently and proved very efficient for interactive image segmentation [6] and image matting [7]. Graph Laplacian methods allow interpretation in terms of MAP estimation in real-valued CRF [8]. A few other interpretations of graph Laplacian methods have been suggested in the literature.

In [6] Grady suggested explanation of using Laplacians for interactive segmentation in terms of random walks. In [9] the use of graph Laplacian for interactive image segmentation was explained in terms of transductive inference. Hein et al. [10] showed that graph Laplacian provides a good approximation for  $s$ -weighted Laplace operator. Therefore, graph Laplacians propose a discrete alternative to the problem of finding a smooth function such that its values in seed pixels are close to the associated labels and it is allowed to vary only on low-density regions of the input space.

In contrast to these works we derive the graph Laplacian by viewing image segmentation as a manifold embedding task. In contrast to [11] we use manifold embedding for semantic image segmentation and not the unsupervised image segmentation. This formulation of semantic image segmentation allows for unsupervised learning of graph Laplacian parameters.

The idea of our learning method is based on the properties of graph Laplacian approximation of Laplace-Beltrami operator studied in [12]. Coifman et al. in [13] proposed a method for automatic selection the kernel bandwidth of graph Laplacian for the problem of optimal manifold embedding. In contrast to [13] we design the method specifically for the task of semantic image segmentation. While the method proposed in [13] aims at choosing one parameter (kernel bandwidth), our method can handle multiple parameters.

### 3. ONE-DIMENSIONAL MANIFOLD EMBEDDING FOR SEMANTIC IMAGE SEGMENTATION

First we discuss the task of manifold embedding and then explain our formulation of image segmentation problem.

#### 3.1 Manifold embedding.

Suppose we have a set of input points  $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^l$ . Let  $d : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}$  be a symmetrical function giving the distance in  $\mathbb{R}^l$ . The optimal manifold embedding task is to find a smooth differentiable function  $f$  that maps the input space  $\mathbb{R}^l$  on the embedded Riemannian manifold  $M$  of dimensionality  $m$  ( $m < l$ ) (see Figure 1, left column). The function  $f$  should preserve distances between the points, such that if  $d(\mathbf{x}_i, \mathbf{x}_j)$  is small, then  $\|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|$  should be small.

Let us focus on the case when the dimension of manifold  $M$  equals one ( $M = \mathbb{R}$ ). Consider two points  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^l$ . They are mapped to  $f(\mathbf{x}), f(\mathbf{y}) \in \mathbb{R}$  respectively. It can be shown [12] that

$$\|f(\mathbf{x}) - f(\mathbf{y})\| \leq d(\mathbf{x}, \mathbf{y}) \|\nabla f(x)\| + o(d(\mathbf{x}, \mathbf{y})), \quad (1)$$

where  $\nabla f(x)$  is the gradient of function  $f(x)$ . Thus we see that  $\nabla f(x)$  provides us with the measure of how far apart  $f$  maps nearby points.

We consider the problem of *initialized one-dimensional manifold embedding* when 1)  $M = \mathbb{R}$  and 2) initial estimates  $y_1, \dots, y_N$  of  $f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)$  in  $\mathbb{R}$  are given. Suppose we know confidences  $c_i \geq 0, i = 1, \dots, N$  that reflect our belief in initial estimates of  $f(\mathbf{x}_i), i = 1, \dots, N$ . Using (1) the problem of initialized one-dimensional manifold embedding can be formulated as minimization the following energy with respect to  $f$

$$E(f) = \sum_i c_i (y_i - f(\mathbf{x}_i))^2 + \int_M \|\nabla f\|^2 dV, \quad (2)$$

where the integral is taken with respect to a standard measure on a Riemannian manifold. The first term in (2) guarantees that corresponding one-dimensional vectors  $f(\mathbf{x}_i)$  are close to their initial estimates  $y_i$ . The second term guarantees that if the points  $\mathbf{x}_i, \mathbf{x}_j$  are close in the input space then their images  $f(\mathbf{x}_i)$  and  $f(\mathbf{x}_j)$  are close in  $M$ .

It follows from the Stokes' theorem that  $\int_M \|\nabla f\|^2 dV = \int_M \Delta_M(f) f dV$ , where  $\Delta_M(f)$  is the Laplace-Beltrami operator. It is a second order differential operator defined as the divergence of the gradient of a function defined on  $M$ .

In many cases finding the mapping  $f$  explicitly is not required. The goal then is to find a set of points  $f(\mathbf{x}_1), \dots, f(\mathbf{x}_N) \in M$  such that represent  $\mathbf{x}_1, \dots, \mathbf{x}_N$ .

#### 3.2 Image segmentation as the manifold embedding problem.

For the sake of clarity first we consider the task of object/background image segmentation. We aim to find real-valued alpha-matting coefficients for each image pixel, the segmentation is then done by thresholding the result. Below we formulate image segmentation problem as the problem of initialized one-dimensional manifold embedding.

Suppose each image pixel is mapped in a feature space  $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^l$ . For example the features can include the spatial coordinates and color of the pixels. Suppose we have defined a

distance function between two pixels  $d : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}$  that tells how likely it is that both pixels belong to object/background.

Suppose that for each image pixel we know the output of some local model  $0 \leq y_i \leq 1, i = 1, \dots, N$  that tells how likely it is that the pixel is a part of the object. Suppose also that we know confidences  $c_i \geq 0, i = 1, \dots, N$  that indicate how much belief we put in the local model.

Our goal is to find real-valued  $f_1, \dots, f_N$  that refine the outputs of local model  $y_1, \dots, y_N$ . We require that  $f_1, \dots, f_N$  lie in the optimally embedded one-dimensional manifold  $M$  and each  $f_i$  corresponds to  $\mathbf{x}_i$ . Therefore the problem of image segmentation reduces to minimization of energy (2).

#### 3.3 Approximation of Laplace-Beltrami operator.

We will now define a graph Laplacian that is an approximation of Laplace-Beltrami operator. Denote weight matrix by  $W : W_{ij} = \exp(-d(\mathbf{x}_i, \mathbf{x}_j))^2$  (in this work we consider only Gaussian kernel). Let  $g_i = \sum_j w_{ij}$  stand for a sum of  $W$  along the  $i$ -th row. Denote diagonal matrix with values  $g_i$  on diagonal by  $D$ . Graph Laplacian is defined as a matrix  $L = W - D$ .

Belkin et al. [14] showed that graph Laplacian  $L$  converges to Laplace-Beltrami operator in the limit  $N \rightarrow \infty$ . In this sense, the graph Laplacian is a numerical machinery for approximating a specific operator on the underlying manifold, by using a finite subset of points.

#### 3.4 Discretization of the problem with graph Laplacian.

Using approximation of Laplace-Beltrami operator by graph Laplacian the problem (2) reduces to minimization of the following energy function with respect to vector  $\mathbf{f} = (f_1, \dots, f_N)$ :

$$E(\mathbf{f}) = \sum_i c_i (f_i - y_i)^2 + \sum_{i,j} w_{ij} (f_i - f_j)^2. \quad (3)$$

The first term in (3) repeats the first term in (2) and the second term in (3) is a discrete approximation of the second term in (2) according to [14]. Minimization of the energy  $E(\mathbf{f})$  can also be interpreted as MAP inference in a real-valued CRF, which are given by the real-valued outputs  $f_1, \dots, f_N$ .

In the matrix form (3) takes the following form:

$$E(\mathbf{f}) = (\mathbf{f} - \mathbf{y})^T C (\mathbf{f} - \mathbf{y}) + \mathbf{f}^T L \mathbf{f}, \quad (4)$$

where  $C$  denotes a square diagonal matrix with  $c_i$  on diagonal and  $\mathbf{y}$  denotes a  $N$ -dimensional vector of initial likelihood scores  $y_i$ . This optimization problem reduces to solving a sparse linear system:

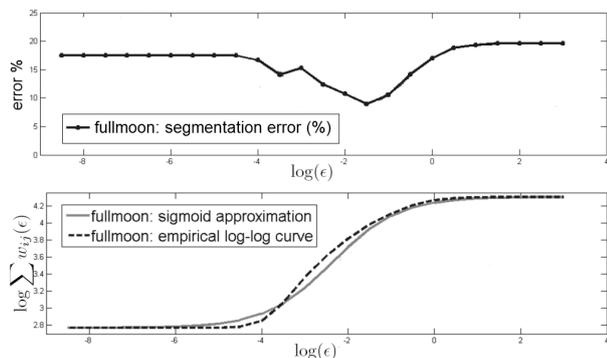
$$(L + C)\mathbf{f} = C\mathbf{y}. \quad (5)$$

The object/background segmentation algorithm then consists in: 1) computing graph Laplacian matrix  $L$ ; 2) solving the sparse linear system (5); 3) thresholding the output.

Described formulation fits both in interactive segmentation scenario and in semantic image segmentation scenario. In case of interactive segmentation confidence values  $c_i$  are infinite for pre-labelled seed points, and 0 for a test points,  $y_i = 1$  for seeds marked as object and equals 0 for background seeds. For semantic segmentation we assume that initial estimates  $y_i$  and confidences  $c_i$  are provided by local models (e.g. appearance model of a specific category).

We notice that instead of pixels we can use image superpixels without making any changes in the algorithm. In the case then superpixels intersect we can average results of (5) for all superpixels that cover image pixel to obtain pixel-wise result.

This framework can be extended to a multi-class segmentation. Let  $K$  denote the number of labels corresponding to object categories. If we solve (5) for each label  $l$  vs all other labels



**Figure 2:** *Top:* segmentation errors for the "fullmoon" image from GrabCut database with respect to  $\log \epsilon$  ( $\alpha$  is fixed). *Bottom:* Dashed line - logarithmic plot for the "fullmoon" image with respect to  $\log \epsilon$  ( $\alpha$  is fixed). The optimal value of  $\epsilon$  is chosen in the point of maximum derivative of the logarithmic plot; Solid line - sigmoid fit of the logarithmic plot.

$1, \dots, l-1, l+1, \dots, K$  and obtain the values  $y_i^{(l)}$  for all image pixels; at the end, an  $i$ -th image pixel is assigned to the label  $l_{max}$ , where  $l_{max} = \arg \max_{l=1, \dots, K} y_i^{(l)}$ .

#### 4. UNSUPERVISED LEARNING OF GRAPH LAPLACIAN PARAMETERS

Suppose that the distance function  $d$  is represented as a weighted sum of metrics  $d_i : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$ ,  $i = 1, \dots, K$ :

$$d(\mathbf{x}_i, \mathbf{x}_j)^2 = \frac{1}{\epsilon} \sum_{k=1}^K \alpha_k d_k(\mathbf{x}_i, \mathbf{x}_j)^2, \quad (6)$$

with fixed  $\alpha_1 = 1$ . Therefore the parameters of graph Laplacian  $\alpha_i, i = 2, \dots, l$  are the weights of features  $\mathbf{x}^k, k = 2, \dots, l$  and the kernel bandwidth  $\epsilon$ . Below we show that optimal value of  $\epsilon$  is determined by the values of  $\alpha_i, i = 2, \dots, l$ .

##### 4.1 Kernel bandwidth $\epsilon$ selection with fixed $\alpha$ .

We start by fixing the parameters  $\alpha_i, i = 2, \dots, l$ . As shown in [13], if we assume that  $L$  provides a good approximation of Laplace-Beltrami operator then the following condition holds:

$$\log \sum_{i,j} w_{ij}(\epsilon) \approx m/2 \log(\epsilon) + \log \left( \frac{N^2 (2\pi)^{m/2}}{\text{vol}(M)} \right), \quad (7)$$

where  $m$  is a dimensionality of corresponding manifold  $M$  and  $w_{ij}$  are the elements of the weight matrix  $W$ .

Consider the *logarithmic plot* of  $\log \sum_{i,j} w_{ij}$  with respect to  $\log \epsilon$ . Figure (2) shows the plot of  $\log \sum_{i,j} w_{ij}$  with respect to  $\log \epsilon$  and  $\log \alpha$  for one image from GrabCut dataset. According to (7) if the approximation is good then the slope of this plot  $\epsilon$  should be about the half dimensionality of corresponding manifold. In the limit  $\epsilon \rightarrow \infty$ ,  $w_{ij} \rightarrow 1$ , so  $\sum_{i,j} w_{ij} \rightarrow N^2$ . On the other hand, as  $\epsilon \rightarrow 0$ ,  $w_{ij} \rightarrow \delta_{ij}$ , so  $\sum_{i,j} w_{ij} \rightarrow N$ . These two limiting values set two asymptotes of the plot and assert that logarithmic plot cannot be linear for all values of  $\epsilon$ .

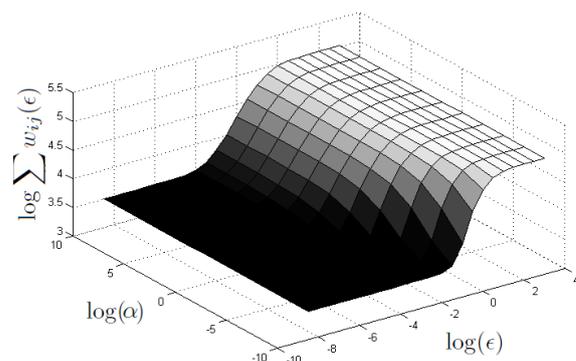
Therefore in order to get better approximation of Laplace-Beltrami operator with  $\alpha_1, \dots, \alpha_K$  fixed we have to choose the value of  $\epsilon$  from the linear region of logarithmic plot. We use the point of maximum derivative as the point of maximum linearity.

##### 4.2 Implementation details

We use the distance function from [9]:

$$\tilde{d}^2(\mathbf{x}_i, \mathbf{x}_j) = \frac{\|r_i - r_j\|^2}{\sigma_r^2} + \frac{\|x_i - x_j\|^2}{\sigma_g^2}, \quad (8)$$

where  $r$  encodes mean RGB color in the superpixel,  $x$  encodes coordinates of the center of the superpixel, parameters of the



**Figure 3:** The plot of  $\log \sum_{i,j} w_{ij}$  with respect to  $\log \epsilon$  and  $\log \alpha$ . The plot shown is shown in (2, bottom) corresponds to the 2-d slice of this 3-d plot for fixed  $\alpha$ . Note that the slope of linear region are not constant for all values of  $\alpha$ . We seek for  $\alpha$  such that the slope in the linear region equals 0.5.

method  $\sigma_r > 0$  and  $\sigma_g > 0$  are the scales of chromatic and geometric neighbourhoods respectively.

This distance function (8) can be rewritten in the form of (6) as:

$$\tilde{d}^2(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\epsilon} (\|r_i - r_j\|^2 + \alpha \|x_i - x_j\|^2), \quad (9)$$

where  $\epsilon = 0.5\sigma_r^2$  and  $\alpha = \sigma_r^2/\sigma_g^2$ . Therefore, the distance function has two parameters  $\epsilon$  and  $\alpha$ .

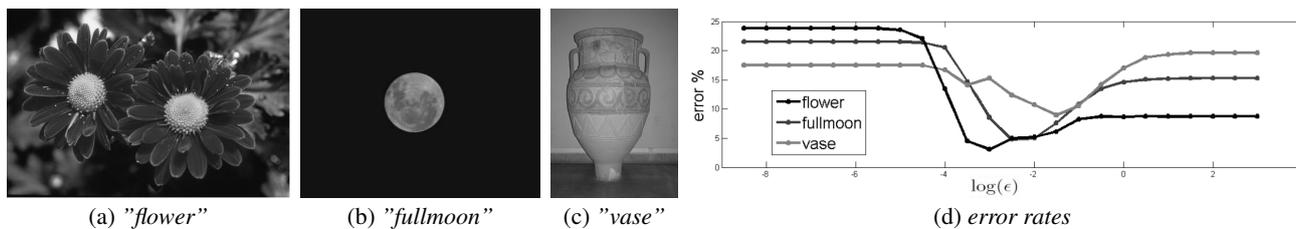
As follows from (7) the slope of the logarithmic curve near optimal value of  $\epsilon$  has to be close to  $m/2$ , where  $m$  is the dimensionality of manifold  $M$ . In our case  $m = 1$ , therefore the slope of the logarithmic plot has to be 0.5. If the plot has different slope in the linear region, this indicates that the second term in (7) is large.

So in the first step of our learning method we should find such  $\alpha$  that the slope of logarithmic plot of  $\log \sum_{i,j} w_{ij}(\epsilon)$  from  $\epsilon$  is equal to 0.5. In the second step we use the sigmoid fit of the logarithmic plot. The shape of logarithmic plot can be approximated with a sigmoid function:  $T(x) = A/(B + \exp(Cx + D)) + E$ . Since the asymptotes of the sigmoid are set by (7) and the slope in the linear region of the sigmoid should be 0.5 the sigmoid has only one free parameter that controls the shift of the sigmoid along horizontal axis. Figure 2 illustrates the choice of  $\epsilon$  according to sigmoid approximation. In our experiments the values of  $\epsilon$  take values as degrees of 10 and the values of  $\alpha$  take values as degrees of 2.

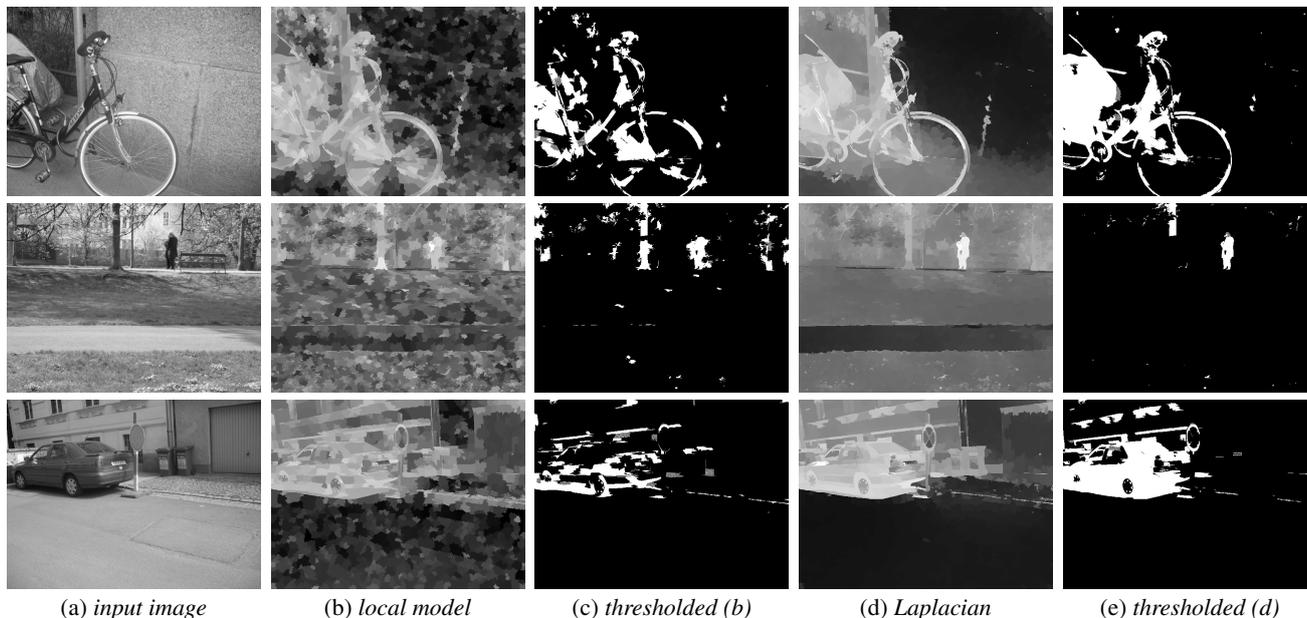
We found empirically that usually the slope of the logarithmic plot is greater than 0.5 for large  $\alpha$  and is less than 0.5 for small  $\alpha$ . In most cases the slope of the logarithmic plot  $S(\alpha)$  is monotonic function of  $\alpha$ . One of the possible explanations of this fact can be the following. Small  $\alpha$  correspond to using only spatial information. This implies that the dimension of manifold where the data lives is 2 and it is difficult to reduce dimensionality further. By decreasing  $\alpha$  we decrease the weight of spatial information in the distance function therefore it gets easier to find the corresponding one-dimensional manifold. On the other hand large  $\alpha$  corresponds to increased weight of color information. Infinite  $\alpha$  corresponds to using color information alone. As long as the color space is three-dimensional and the color distribution of object and background is complex it is difficult to embed one-dimensional manifold in the input points. Example of the logarithmic plot with respect to both  $\epsilon$  and  $\alpha$  is shown in Figure 3.

## 5. EXPERIMENTS

For the experiments we used GrabCut, Graz and Pascal 2007 datasets. In all experiments graph Laplacian operated with superpixels produced by image over-segmentation methods. Each



**Figure 4:** Segmentation errors depending on graph Laplacian parameter  $\epsilon$  on three images from GrabCut dataset. Note that minimal error is achieved on different values of  $\epsilon$  for different images.



**Figure 5:** Results of SVM and graph Laplacian method for images from Graz dataset. (a) - input images of "bike", "person" and "cars" classes; (b) - real-valued output from local SVM model; (c) - results of thresholding the SVM outputs; (d) - real-valued output of graph Laplacian using SVM as a local model with the parameters learnt by our method; (e) - thresholded output of our method. Note how graph Laplacian refines the output from SVM. It doesn't oversmooth the result and preserves fine details like the wheel of the bike and the small figure of the person.

superpixel was linked with a fixed number of its nearest neighbours, and the distances to other superpixels were assumed infinite. For all experiments we used confidences that are a linear function of the outputs of local appearance models  $c_i = 0.5(1 - |p_i - 0.5|)$ .

### 5.1 GrabCut image database.

GrabCut image database contains 50 images of different objects<sup>1</sup>. In the experiments we used the set of superpixels which is the union of oversegmentations provided by Colour Structure Code and Watershed segmentation methods.

Figure 4 (d) shows the error on 3 different images from GrabCut database with respect to  $\log \epsilon$  ( $\alpha$  is fixed). Depending on the choice of  $\epsilon$  one can get different values of errors and for each image, and the optimal values of  $\epsilon$  are different for different images.

We measured performance on GrabCut dataset according to standard metric [1]. We compared three versions of graph Laplacian. First, we chose the best parameters for each image individually by validation on the same image in order to obtain the top bound on performance of graph Laplacian. The resulting error rate is 8.7%. In the second experiment we chose single set of parameters for the whole dataset by validation on the test dataset. This corresponds to upper bound on performance of the method with

fixed parameters. The resulting error rate is  $9.9 \pm 4.3\%$ . Finally we evaluated the performance of graph Laplacian with the parameters learnt individually for each image by our method. The resulting error rate is  $10.2 \pm 4.0\%$ . The results of graph Laplacian with learnt parameters is very close to the upper bound of performance of graph Laplacian with fixed parameters. Notably, the standard deviation of errors obtained with the parameters learnt individually for each image is smaller than of Laplacian with fixed parameters.

The learning phase took from 0.5 seconds to 3 seconds, solving linear system 5 took from 0.05 second to 0.5 seconds depending on the number of superpixels in the image (the total number of superpixels varied from 500 to 30000).

### 5.2 Graz image dataset.

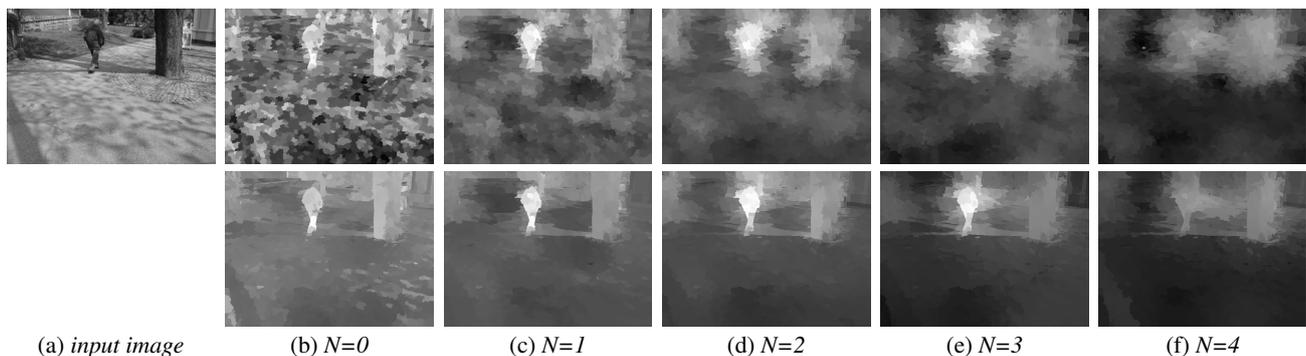
Graz dataset<sup>2</sup> contains 1096 images of three classes: "person", "bike" and "car". In our experiments we solved a separate binary segmentation problem for each category. To measure the quality of segmentation we used a standard metric - percent of incorrectly classified pixels in the image.

In our experiments we used an open-source VIBlocks toolbox<sup>3</sup>, which implements the method described in [15]. We chose it for comparison for the following reasons. First, it allows using dif-

<sup>1</sup>available at <http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm>

<sup>2</sup>available at <http://www.emt.tugraz.at>

<sup>3</sup>code available at <http://vlblocks.org/index.html>



**Figure 6:** Results of using different local models. The first row shows real-valued output of local appearance models. The second row shows results of our method. Parameter  $N$  sets the size of superpixel neighborhood in the local model. The effect of using graph Laplacian is better visible for smaller  $N$ .

	N=0			N=1			N=2			N=3			N=4		
	cars	bike	pers	cars	bike	pers	cars	bike	pers	cars	bike	pers	cars	bike	pers
<i>SVM</i>	41.9	56.5	49.4	59.6	66.9	63.6	68.0	69.2	66.6	69.4	70.7	65.2	66.5	71.9	63.6
+ <i>CRF</i>	43.0	57.7	49.3	60.2	67.1	63.9	70.1	70.2	66.9	70.7	71.0	65.4	68.8	72.2	64.2
+ <i>Laplacian (valid.GrabCut)</i>	50.0	60.1	56.0	65.5	68.7	68.5	71.6	70.8	70.8	72.2	72.0	69.5	70.0	<b>73.2</b>	67.3
+ <i>Laplacian (valid.test set)</i>	<b>56.6</b>	<b>63.3</b>	<b>59.1</b>	66.3	68.4	68.8	71.9	70.4	70.4	72.6	71.2	69.4	70.8	72.2	68.0
+ <i>Laplacian (learnt)</i>	54.2	60.9	58.5	65.1	66.8	<b>69.4</b>	<b>72.0</b>	69.5	<b>71.3</b>	<b>73.3</b>	70.3	<b>70.2</b>	<b>71.4</b>	71.5	<b>68.9</b>

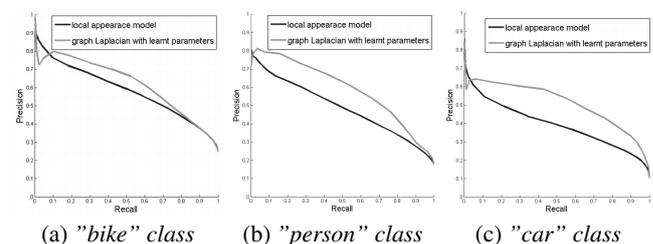
**Table 1:** Performance on Graz dataset at equal precision and recall rates for "cars", "bike" and "person" classes. First row: local appearance model (from VIBlocks toolbox). Second row: result of applying discrete CRF with graph cut inference (from VIBlocks toolbox). Third row: graph Laplacian with parameters validated on GrabCut dataset. Fourth row: graph Laplacian with parameters validated on the test set. Fifth row: graph Laplacian with parameters learnt individually for each image. For each appearance model used in our experiments (we varied the number of neighboring regions as in [15]) the best result is shown in **bold font**. Underlined are the best overall results.

ferent local appearance models. The method has a parameter  $N$  meaning number of neighbouring superpixels which features are used for classification of each particular superpixel. So we report performance metrics for different values of  $N$  to illustrate the performance of proposed graph Laplacian framework applied to different local models. Second, the toolbox includes implementation of discrete CRF with graph-cut inference, which we use for comparison. Note, this CRF model uses similar types of features (color and spatial coordinates of superpixels) to those used in our graph Laplacian. We used the same over-segmentation and the same local appearance model based on SVM as [15]. To obtain initial estimates  $y_i$  for graph Laplacian framework we scaled SVM outputs to  $[0, 1]$  interval for each image.

In the first experiment the parameters  $\epsilon$  and  $\alpha$  were validated on the GrabCut dataset. In the second experiment we validated the parameters on the test set. In the third experiment we used our unsupervised learning method for choosing the parameters individually for each image. We also compared with VIBlocks implementation of CRF with graph-cut inference. The strategy for choosing internal parameters of CRF was the same as in [15].

Table 1 contains results of the comparison. Our unsupervised learning gives results comparable to upper bound on performance of graph Laplacian with fixed parameters from the second experiment. The value of performance gain compared to local appearance model differs for different values of parameter  $N$ . The smaller  $N$  is the smaller neighborhood is considered by low-level model, and the more significant is the gain in performance attained by both CRF and graph Laplacian. The gain in performance of graph Laplacian is almost uniformly higher than the performance gain obtained by discrete CRF. Note that the gain achieved by graph Laplacian is several times higher than the one achieved by discrete CRF for  $N = 0$ . Figure 7 shows precision-recall curves for local appearance models and for our method to illustrate the gain in performance due to graph Laplacian.

Figure 5 shows results provided by local appearance model (SVM) and corresponding results of using graph Laplacian with



**Figure 7:** Precision-recall curves for "bike", "person" and "car" classes of Graz dataset. *Blue curves* - local appearance model ( $N=0$ ); *Green curves* - graph Laplacian with learnt parameters.

learnt parameters. Figure 6 shows how the results vary for different local models.

The running time on Graz dataset is the following: learning phase takes about 0.2 seconds on average, solving of linear system 5 takes about 0.02 seconds on average.

### 5.3 Pascal 2007 image dataset.

Pascal 2007 dataset <sup>4</sup> contains 21 classes. Again in this experiment we use local models from VIBlocks toolbox trained with parameters as in [15]. We compare our graph Laplacian method with unsupervised learning to the discrete CRF implemented in VIBlocks toolbox. The training and testing split is defined in the challenge. We train local model on the train set and choose the parameters of discrete CRF on the validation set. We do not use the validation set in the experiment with our graph Laplacian method and use our unsupervised learning method for choosing Laplacian parameters for each particular image.

The table 5.2 shows the results of comparison to using local

<sup>4</sup>available at <http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/>

	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dinningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tv monitor	Average	Pixels %
<i>Pantofaru et al. [16]</i>	59	27	1	8	2	1	32	14	14	4	8	32	9	24	15	81	11	<b>26</b>	1	28	17	20	-
<i>Shotton et al. [17]</i>	33	46	5	14	11	14	34	8	6	3	10	39	40	28	23	32	19	19	8	24	9	20	-
<i>Fulkerson et al. [15]</i>	56	26	<b>29</b>	<b>19</b>	16	3	<b>42</b>	44	<b>56</b>	23	6	11	62	16	68	46	16	10	21	52	<b>40</b>	<b>32</b>	51
<i>local model (N=0)</i>	16	17	8	9	13	8	9	9	29	15	9	12	12	7	13	5	16	16	26	14	21	14	15
<i>+discrete CRF (N=0)</i>	17	18	8	9	<b>21</b>	8	9	8	28	14	9	13	12	7	13	5	16	16	26	14	22	14	16
<i>+Ours(learnt) (N=0)</i>	23	20	26	12	18	12	17	14	49	17	1	23	16	11	50	4	<b>42</b>	20	<b>44</b>	30	33	23	22
<i>local model (N=1)</i>	24	13	18	13	9	12	14	24	35	16	9	11	28	15	36	23	14	21	20	35	27	21	38
<i>+discrete CRF (N=1)</i>	42	6	16	9	6	5	11	14	56	19	4	11	16	16	55	36	24	16	8	56	21	21	38
<i>+Ours(learnt) (N=1)</i>	33	11	18	14	10	13	16	26	51	16	7	9	35	<b>22</b>	67	29	31	25	16	<b>60</b>	33	26	32
<i>local model (N=2)</i>	39	10	22	15	11	12	18	36	44	23	8	11	33	15	53	37	17	17	16	36	28	24	36
<i>+discrete CRF (N=2)</i>	58	9	25	14	6	3	20	38	54	<b>27</b>	12	10	31	7	59	44	12	17	13	43	27	25	50
<i>+Ours(learnt) (N=2)</i>	52	9	24	17	8	12	19	41	55	21	10	10	<b>37</b>	14	68	42	14	12	14	51	32	27	46
<i>local model (N=3)</i>	52	13	14	18	8	5	23	38	45	17	7	10	30	21	63	50	17	20	19	43	23	25	46
<i>+discrete CRF (N=3)</i>	65	10	14	15	5	2	24	40	60	13	6	8	24	19	68	55	18	19	16	46	26	26	56
<i>+Ours(learnt) (N=3)</i>	63	11	15	17	7	2	22	40	60	13	5	10	32	20	<b>71</b>	57	13	17	14	48	26	27	55
<i>local model (N=4)</i>	59	6	15	18	4	0	25	44	46	17	3	4	24	20	62	56	15	14	13	38	33	25	51
<i>+discrete CRF (N=4)</i>	63	8	15	18	4	0	26	<b>46</b>	48	17	2	4	25	20	64	58	12	14	12	38	34	25	54
<i>+Ours(learnt) (N=4)</i>	69	5	11	<b>19</b>	4	0	28	45	55	13	2	3	18	18	69	<b>60</b>	7	13	7	44	36	25	<b>59</b>

Table 2: Results on Pascal 2007 dataset. Best result for each category is shown in bold.

model alone, discrete CRF and our graph Laplacian method with unsupervised learning. For comparison we also reproduce results from [17], [16] and [15]. On this dataset, adding graph Laplacian improves the results significantly compared to using local model alone, and provides a consistent boost for the accuracy as well.

## 6. CONCLUSION

We presented an semantic segmentation framework based on graph Laplacian approximation of manifold embedding problem. The main contribution of this work is a method for choosing internal parameters of graph Laplacian in a fully unsupervised manner individually for each test image. Proposed unsupervised learning method has a low computational cost and shows better performance compared to discrete CRF with graph-cut inference.

## 7. REFERENCES

- [1] Yuri Boykov and Marie-Pierre Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *ICCV*, 2001, vol. 1, pp. 105–112.
- [2] Pushmeet Kohli, Lubor Ladicky, and Philip Torr, "Robust higher order potentials for enforcing label consistency," in *CVPR*, 2008.
- [3] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip Torr, "Graph cut based inference with co-occurrence statistics," in *ECCV*, 2010.
- [4] Martin Szummer, Pushmeet Kohli, and Derek Hoiem, "Learning crfs using graph cuts," in *ECCV*, 2008.
- [5] Sebastian Nowozin, Peter V. Gehler, and Christoph H. Lampert, "On parameter learning in crf-based approaches to object class image segmentation," in *Proceedings of the 11th European conference on Computer vision: Part VI*, 2010, ECCV'10, pp. 98–111.
- [6] Leo Grady, "Random walks for image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.
- [7] Anat Levin, Dani Lischinski, and Yair Weiss, "A closed form solution to natural image matting," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2008.
- [8] Dheeraj Singaraju, Leo Grady, and Ren Vidal, "P-brush: Continuous valued mrfs with normed pairwise distributions for image segmentation," in *CVPR*, 2009.
- [9] Olivier Duchenne, Jean-Yves Audibert, Renaud Keriven, Jean Ponce, and Florent Segonne, "Segmentation by transduction," in *CVPR*, 2008.
- [10] M. Hein, J.-Y. Audibert, and U. von Luxburg, "From graphs to manifolds - weak and strong pointwise consistency of graph laplacians," *ArXiv Preprint, Journal of Machine Learning Research, forthcoming*, 2006.
- [11] H.B. Zhou and Q.A. Cheng, "O(n) implicit subspace embedding for unsupervised multi-scale image segmentation," 2011, pp. 2209–2215.
- [12] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computaion*, vol. 15, pp. 1373–1396, 2003.
- [13] Ronald R. Coifman, Yoel Shkolnisky, Fred J. Sigworth, and Amit Singer, "Graph laplacian tomography from unknown random projections," *IEEE Trans. on Image Processing*.
- [14] Mikhail Belkin and Partha Niyogi, "Towards a theoretical foundation for laplacian-based manifold methods," in *COLT*, 2005, pp. 486–500.
- [15] Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *ICCV*, 2009.
- [16] Caroline Pantofaru, Cordelia Schmid, and Martial Hebert, "Object recognition by integrating multiple image segmentations," in *ECCV*, 2008.
- [17] Jamie Shotton, Matthew Johnson, and Roberto Cipolla, "Semantic texton forests for image categorization and segmentation," 2008.