# P-Brush: Continuous Valued MRFs with Normed Pairwise Distributions for Image Segmentation *

Dheeraj Singaraju
Johns Hopkins University
Baltimore, MD
dheeraj@cis.jhu.edu

Leo Grady
Siemens Corporate Research
Princeton, NJ
leo.grady@siemens.com

René Vidal
Johns Hopkins University
Baltimore, MD
rvidal@cis.jhu.edu

## Abstract

*Interactive image segmentation traditionally involves the use of algorithms such as Graph Cuts or Random Walker. Common concerns with using Graph Cuts are metrication artifacts (blockiness) and the shrinking bias (bias towards shorter boundaries). The Random Walker avoids these problems, but suffers from the proximity bias (sensitivity to location of pixels labeled by the user). In this work, we introduce a new family of segmentation algorithms that includes Graph Cuts and Random Walker as special cases. We explore image segmentation using continuous-valued Markov Random Fields (MRFs) with probability distributions following the p-norm of the difference between configurations of neighboring sites. For p=1 these MRFs may be interpreted as the standard binary MRF used by Graph Cuts, while for p=2 these MRFs may be viewed as Gaussian MRFs employed by the Random Walker algorithm. By allowing the probability distribution for neighboring sites to take any arbitrary p-norm ($p \geq 1$), we pave the path for hybrid extensions of these algorithms. Experiments show that the use of a fractional $p$ ($1 < p < 2$) can be used to resolve the aforementioned drawbacks of these algorithms.*

## 1. Introduction

Object segmentation is an important part of scene understanding and interpretation, which involves finding the regions of the image that correspond to an *object*. Alternatively, it can be posed as the problem of finding the boundaries of the object in the image. Object segmentation is an intrinsically ill-posed problem, because a scene could contain several objects. Hence, determining what constitutes an object of interest could be ambigious. Such ambiguities can be resolved by allowing the user to interact with the algorithm and provide information about the object of interest.

In this work, we restrict our analysis to a subset of interactive segmentation algorithms that input a partial labeling of the pixels and produce a complete labeling of the pixels. More specifically, we consider *seeded algorithms* (i.e., a "scribble interface") that input a set of pixels as belonging to *object* and a disjoint set of pixels as belonging to *background*. The goal of the algorithm is to output a labeling of every pixel into one of these two categories.

**Prior work.** A comprehensive review of the literature on seeded segmentation algorithms is beyond the scope of this paper. We will focus our review on the relevant, recent seeded segmentation algorithms of Graph Cuts, Random Walker and shortest path. These methods typically construct a graph such that each node in the graph corresponds to a pixel in the image. Neighboring nodes are connected by edges that are assigned non-negative weights. These weights are decreasing functions of the difference between the intensities, colors or other features of the nodes (pixels) connected by the edges. The labels are then estimated by minimizing an energy function defined on this graph, subject to the constraints enforced by the scribble.

The Graph Cuts algorithm initially introduced in [5], estimates the labeling by finding the minimum cut between the foreground and background seeds via a maximum flow computation. Two concerns in the literature about the original Graph Cuts algorithm are metrication error (blockiness) and the shrinking bias (bias towards shorter boundaries). Subsequent work has addressed the metrication error [6, 3, 22] and the shrinking bias [3, 23], but at the expense of additional parameters, memory or computation.

The Random Walker algorithm introduced in [10], assigns each unlabeled pixel to the seed for which there is a minimum diffusion distance. This segmentation principle has also been interpreted as an interactive version of normalized cuts [18, 11]. The benefit of using the diffusion distance for labeling is that weak or noisy boundaries do not cause bleeding of the segmentation. Also, the use of this distance does not result in a shrinking bias. However, the negative aspect in comparison to Graph Cuts is that the segmentation boundary is more strongly affected by the seed location [20]. We term this effect as the *proximity bias*.

*This work was partly supported by JHU startup funds, by grants ONR N00014-05-10836, ONR N00014-09-1-0084, NSF CAREER 0447739 and ONR Young Investor Award.

Shortest path algorithms assigns each pixel to the foreground if there is a shorter path on the weighted graph from that pixel to a foreground seed than to any background seed. This approach was recently popularized by [4], but variants of this idea have appeared in several sources [8, 1, 2]. The primary advantage of the shortest paths algorithm is speed and prevention of a shrinking bias. However, the disadvantage of such methods is that a single good path is sufficient for connectivity. Consequently, as shown in [20], these algorithms exhibit much stronger proximity bias than the Random Walker and Graph Cuts, and are more likely to leak through weak boundaries. Shortest path algorithms also exhibit metrication artifacts on a 4-connected lattice.

The aforementioned segmentation strategies were recently shown to be special cases of a single more general algorithm [20]. This general algorithm attempts to find a potential function with a minimum spatial gradient, subject to Dirichlet boundary conditions given at the seeds. The algorithm proceeds by minimizing the $p$-norm of the gradient and it was shown that $p = 1$ gives the Graph Cuts algorithm, $p = 2$ gives the Random Walker algorithm and $p = \infty$ gives the shortest path algorithm. In [20], it was assumed that each of these three values for $p$ could be minimized using specialized algorithms (e.g., max-flow, linear equation solver and Dijsktra's algorithm). This assumption precluded the ability to merge algorithms or employ the generalized algorithm with a $p$ value not equal to 1, 2 or $\infty$, since no algorithm is given for minimizing an arbitrary $p$-norm. However, when $p \geq 1$, the objective energy is convex and we therefore expect to find a global optimum.

**Paper contributions.** In this paper, we explore minimizing the cost function of [20] for $1 < p < 2$, to see if an amenable trade off can be achieved between Graph Cuts and the Random Walker. We focus on these two algorithms since their properties complement each other, as discussed above. Specifically, we propose a new cost function for interactive image segmentation and an efficient solver for minimizing this function. The proposed cost function generalizes that in [20] by allowing the use of different norms for the spatial gradients at different nodes in the constructed graph. Hence, our framework admits hybrid combinations of algorithms such as Graph Cuts and the Random Walker.

We use *Iterative Reweighted Least Squares* (IRLS) techniques to find the global minimizer of our proposed cost function by solving a series of $\ell_2$ optimizations. We show that IRLS gives an elegant yet effective solution for image segmentation. Consequently, we provide new solvers for the algorithms of [5] and [20]. Finally, we present experiments on synthetic images and an extensive database of 50 real images. We illustrate the advantage of our proposed framework, and show that use of a hybrid algorithm allows us to keep the benefits of both algorithms while simultaneously addressing their individual drawbacks.

## 2. Continuous Valued MRFs for Image Segmentation

In this section, we formalize the construction of the combinatorial graph and define an energy function on this graph, the minimization of which gives us the required segmentation. Finally, we show how our framework corresponds to finding the MAP estimate of a continuous valued MRF.

We define a weighted undirected combinatorial graph $\mathcal{G}$ as a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $i \in \mathcal{V}$ and edges $\boldsymbol{e}_{ij} \in \mathcal{E}$. An edge that spans two vertices $i$ and $j$ is denoted by $\boldsymbol{e}_{ij}$. We assign to each edge $\boldsymbol{e}_{ij}$ a non-negative value $w_{ij}$ that is referred to as its weight and we assume that $\forall \boldsymbol{e}_{ij} \in \mathcal{E} : w_{ij} = w_{ji}$. The neighborhood of a node $i$ is given by all the nodes that it shares an edge with and is denoted by $\mathcal{N}_i$. The nodes on the graph typically correspond to pixels in the image and we employ a 4 connected grid in all our experiments. Also, we focus on intensity based segmentation and consequently define the edge weights as $w_{ij} = \exp(-\beta \|I_i - I_j\|^2)$, where $\beta > 0$ and $I_i$ is the grayscale intensity or RGB color of the node $i$. One can also use other suitable node-specific image features in order to to define the edge weights.

Our algorithm requires that the foreground/background seed nodes be given interactively or automatically. The set $\mathcal{M} \subset \mathcal{V}$ contains the locations of the nodes marked as seeds and the set $\mathcal{U} \subset \mathcal{V}$ contains the locations of the unmarked nodes. We split the set $\mathcal{M}$ into the sets $\mathcal{O} \subset \mathcal{M}$ and $\mathcal{B} \subset \mathcal{M}$ that contain the locations of the seeds for the object and the background, respectively. By construction, (a) $\mathcal{M} \cap \mathcal{U} = \emptyset$, (b) $\mathcal{M} \cup \mathcal{U} = \mathcal{V}$, (c) $\mathcal{O} \cap \mathcal{B} = \emptyset$, and (d) $\mathcal{O} \cup \mathcal{B} = \mathcal{M}$.

Each node $i \in \mathcal{V}$ is associated with a membership $x_i \in [0, 1]$. The nodes labelled by the user are assigned membership as $\forall i \in \mathcal{O}, x_i = 1$ and $\forall i \in \mathcal{B}, x_i = 0$. For the sake of convenience, we define a membership vector $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{V}|}$, whose $i^{\text{th}}$ entry is given by $x_i$. We also define vectors $\boldsymbol{x}_U = \{x_i\}_{i \in \mathcal{U}}$ and $\boldsymbol{x}_M = \{x_i\}_{i \in \mathcal{M}}$ that contain the memberships of the unmarked nodes and the marked nodes, respectively. Now, for the purpose of image segmentation, we define an energy $E_{\mathcal{P}}(\boldsymbol{x})$ on the graph $\mathcal{G}$ as

$$E_{\mathcal{P}}(\boldsymbol{x}) = \sum_{\boldsymbol{e}_{ij} \in \mathcal{E}} w_{ij} |x_i - x_j|^{p_{ij}}, \tag{1}$$

where $\forall \boldsymbol{e}_{ij} \in \mathcal{E}, 1 \leq p_{ij} < \infty$ and $\mathcal{P} = \{p_{ij}\}$. Note that this energy function contains terms for pairwise interaction between nodes. However, *unary terms* can also be accounted for by introducing auxiliary nodes in the graph for the seed nodes and modeling the unary terms as pairwise interactions with these auxiliary nodes. Now given $E_{\mathcal{P}}(\boldsymbol{x})$, we propose to estimate the memberships of the pixels as

$$\boldsymbol{x} = \underset{\boldsymbol{x}}{\operatorname{argmin}}\ E_{\mathcal{P}}(\boldsymbol{x}) \text{ s.t. } x_i = 1, \text{ if } i \in \mathcal{O} \text{ and } x_i = 0, \text{ if } i \in \mathcal{B}. \tag{2}$$

Observe that if $\forall \boldsymbol{e}_{ij} \in \mathcal{E}, p_{ij} = p \in [1, \infty)$, the function $E_{\mathcal{P}}(\boldsymbol{x})$ reduces to the cost function proposed by [20]. Consequently, our framework is a true generalization of [20], where we allow different norms for different edges rather than a common norm for all the edges[1]. Since we are considering the case of arbitrary $p \geq 1$ values, we will refer to the algorithm that solves (2) as the ***p*-brush** algorithm.

It is useful to interpret the potential field produced by the $p$-brush algorithm as the MAP estimate of a Markov Random Field with continuous valued variables, which are given by the memberships $x_i$ of the nodes in the graph. The variables at the marked pixels are treated as observed variables such that $\forall i \in \mathcal{O}, x_i = 1$ and $\forall i \in \mathcal{B}, x_i = 0$ We define the probability for any configuration $\boldsymbol{x}$ to be given as

$$p(\boldsymbol{x}) = \frac{1}{Z} \exp\left( - \sum_{e_{ij} \in E} w_{ij} |x_i - x_j|^{p_{ij}} \right), \qquad (3)$$

where $Z$ is the partition constant. Since $Z$ does not depend on $\boldsymbol{x}$, the MAP estimate of this MRF is given by (2). It is interesting that Graph Cuts may now also be viewed as optimizing a continuous valued MRF, as opposed to the traditional view of minimization of a *binary* MRF [9, 5]. Also, for $p_{ij} = 2$, our formulation corresponds to optimizing a Gaussian MRF [21], as done by the Random Walker.

Now, note that for the MAP estimate, the membership $x_i$ of an unmarked node $i \in \mathcal{U}$ takes the value

$$x_i = \underset{x}{\operatorname{argmin}} \left( \sum_{j \in \mathcal{N}_i} w_{ij} |x - x_j|^{p_{ij}} \right). \qquad (4)$$

This allows us to appreciate the relationship between variables at one node with the variables at neighboring nodes. The fact that the configuration of a node is dependent only on the configuration of the neighboring nodes, implies that our formulation satisfies the Markovian property.

Now, if we assume that $\forall e_{ij}, w_{ij} = 1$ and $p_{ij} = p$ then the membership $x_i$ of an unmarked node $i$ will take the $p$-average of its neighbors. Specifically, when $p = 0, 1, 2$ or $\infty$, then $x_i$ will take the mode, median, mean or minimax value of its neighbors, respectively. If the weights are nonuniform, then the weighted equivalents of these quantities will be determined by each $x_i$. The interpretation of Graph Cuts, Random Walker and shortest paths in terms of the metric average of the potential function of its neighbor allows us to interpret the shrinking bias as the outlier removal of the median (where seeds comprise the outliers) and proximity bias by the oversmoothness of the mean and minimax norms. Therefore, fractional values of $p$ (such as $p = 1.5$) may be interpreted as the use of a *robust* norm to prevent oversmoothness while preventing the shrinking bias by allowing the outliers (seeds) to have stronger influence.

[1]Note that in contrast to [20], we follow the notation of using $p_{ij}$ for the exponent in the term $|x_i - x_j|^{p_{ij}}$, as opposed to $q$.

## 3. Properties of the Solutions

We note that by construction $E_{\mathcal{P}}(\boldsymbol{x})$ is a convex continuous function of $\boldsymbol{x}$, since $\forall \boldsymbol{e}_{ij} \in \mathcal{E}, p_{ij} \geq 1$. Therefore, any solution $\boldsymbol{x}^*$ of (2) must be a global minimizer of $E_{\mathcal{P}}(\boldsymbol{x})$. In this section, we characterize the nature of these minimizers. Specifically, we generalize the maximum (minimum) principle governing elliptical equations in physics to show that $\boldsymbol{x}^*$ must satisfy the property $\forall i \in \mathcal{U}, x_i^* \in [0, 1]$.

**Theorem 1 (Intermediate Value Theorem)** *Let $\boldsymbol{x}^*$ be a solution to the minimization problem in* (2)*. The membership $x_i^*$ of each unmarked node $i \in \mathcal{U}$ is bounded by the maximum and minimum of the memberships $x_j^*$ of the neighboring nodes $j \in \mathcal{N}_i$.*

*Proof.* We prove the result by contradiction. Assume that for the given solution $\boldsymbol{x}^*$, there exists an unmarked node $i \in \mathcal{U}$ such that $\forall j \in \mathcal{N}_i, x_i^* > x_j^*$. We define a new vector $\boldsymbol{y}^* \in \mathbb{R}^{|\mathcal{V}|}$ as $y_k^* = x_k^*$, if $k \neq i$ and $y_i^* = \max_{j \in \mathcal{N}_i} x_j$. Now,

$$E_{\mathcal{P}}(\boldsymbol{y}^*) - E_{\mathcal{P}}(\boldsymbol{x}^*) = \sum_{j \in \mathcal{N}_i} w_{ij} \left( (y_i^* - x_j^*)^{p_{ij}} - (x_i^* - x_j^*)^{p_{ij}} \right) < 0,$$

since $y_i^* < x_i^*$ and $\forall j \in \mathcal{N}_i, p_{ij} \geq 1$. This contradicts the fact that $\boldsymbol{x}^*$ is a global minimizer of $E_{\mathcal{P}}(\boldsymbol{x})$, hence proving that the value of $x_i^*$ is bounded above by the maximum of the memberships $x_j^*$ of the neighboring nodes $j \in \mathcal{N}_i$. By a similar argument, we can prove the lower bound of $x_i^*$. ∎

Note that Theorem 1 provides local bounds on the membership of a node based on the memberships of the neighboring nodes. However, Corollary 1 gives a more powerful result which helps us determine global bounds on the memberships of all the nodes in the graph.

**Corollary 1 (Extremum Value Property)** *Let $\boldsymbol{x}^*$ be a solution to the minimization problem in* (2)*. The membership $x_i^*$ of each node $i \in \mathcal{V}$ satisfies the property:*

$$\forall i \in \mathcal{V}, \min_{j \in \mathcal{M}} x_j^* \leq x_i^* \leq \max_{j \in \mathcal{M}} x_j^*. \qquad (5)$$

*Proof.* Let us assume that the maximum of the memberships $\{x_i^*\}$ is achieved at an unmarked node $j \in \mathcal{U}$, *i.e.* $\forall i \in \mathcal{V}, x_j^* \geq x_i^*$. We now use the following recursive construction to sketch our proof.

1. Define a set $M = \{j\}$. This set will be updated recursively to include the nodes $i \in \mathcal{V}$ that satisfy $x_i^* = x_j^*$.

2. Check if a marked node $k$ is contained in $M$. The proof is complete if $k \in M$, for some $k \in \mathcal{M}$, since we have shown that the maximum value of the memberships is achieved at a marked node. If not, proceed to step 3.

3. Define a set $\mathcal{N}_M$ as the neighborhood of all the nodes $i$ contained in $M$, *i.e.* $\mathcal{N}_M = (\cup_{i\in M}\mathcal{N}_i)\setminus M$. There exists at least one node $k \in \mathcal{N}_M$ such that $x_k^* = x_j^*$. If not, then by an argument similar to the proof of Theorem 1, we can construct a new vector $\boldsymbol{y}^*$ from $\boldsymbol{x}^*$, such that $\forall k \in M, y_k^* = \max_{i\in\mathcal{N}_M} x_i^* < x_j^*$. Then $E_{\mathcal{P}}(\boldsymbol{y}^*) < E_{\mathcal{P}}(\boldsymbol{x}^*)$ contradicts the hypothesis that $\boldsymbol{x}^*$ is a global minimizer of the cost function $E_{\mathcal{P}}(\boldsymbol{x})$.

4. Append the set $M$ with the node $k$ obtained from step 2 as $M = M \cup \{k\}$, and go to step 2.

Note that since $\forall \boldsymbol{e}_{ij} \in \mathcal{E}, w_{ij} > 0$, we can find a path between any pair of nodes in the graph defined using a 4-connected neighborhood. Therefore, we are assured of that fact that $M \cap \mathcal{M} \neq \emptyset$ at some step of our described search strategy. A negation of this statement would imply that the search terminates in a set $M \subset \mathcal{U}$ that satisfies (a) $(\mathcal{U}\setminus M)\cap \mathcal{N}_M = \emptyset$, since otherwise an unmarked node from this set would be appended in step 3, and (b) $\mathcal{U} \cap \mathcal{N}_M = \emptyset$, since we assumed our search strategy terminated without finding any marked node in $M$. These conditions imply that the sets $M$ and $(U \setminus M) \cup \mathcal{M}$ are disconnected, contradicting the fully connectedness hypothesis. Similarly, we can prove the result for the minimum value of the memberships too. ∎

Since by construction we have $\forall i \in \mathcal{O}, x_i = 1$ and $\forall i \in \mathcal{B}, x_i = 0$, we conclude from Corollary 1 that the membership $x_i$ of each unmarked node $i \in \mathcal{U}$ takes values in $[0, 1]$. As a result, the set of solutions for the potentials at the unmarked nodes $\boldsymbol{x}_{\mathcal{U}}$ is $[0, 1]^{|\mathcal{U}|}$ which is a compact and convex set. This result coupled with the fact that the energy function $E_{\mathcal{P}}(\boldsymbol{x})$ is convex in $\boldsymbol{x}$, allows us to adopt any descent algorithm in order to estimate the minimizer of $E_{\mathcal{P}}(\boldsymbol{x})$. Since the estimated membership $x_i$ of each unmarked node $i \in \mathcal{U}$ lies in $[0, 1]$, we estimate the segmentation by placing a decision threshold at $x = 0.5$.

Now, we derive a result which will assist us in the convergence analysis of our proposed iterative solver for $\boldsymbol{x}^*$.

**Theorem 2 (Right continuity in the norms $\mathcal{P} = \{p_{ij}\}$)** *Define $\boldsymbol{x}^*_{\mathcal{P}+\epsilon}$ as a minimizer of $E_{\mathcal{P}+\epsilon}(\boldsymbol{x})$, where $\mathcal{P} + \epsilon = \{p_{ij} + \epsilon\}_{\boldsymbol{e}_{ij}\in E}, \epsilon \geq 0$. Then $\boldsymbol{x}^*_{\mathcal{P}+\epsilon}$ is right continuous in the entries of $\mathcal{P}$,* i.e.

$$\lim_{\epsilon\to 0} \boldsymbol{x}^*_{\mathcal{P}+\epsilon} = \boldsymbol{x}^*_{\mathcal{P}}. \tag{6}$$

*Proof.* By definition, we have that $E_{\mathcal{P}}(\boldsymbol{x}^*_{\mathcal{P}}) \leq E_{\mathcal{P}}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon})$ and $E_{\mathcal{P}+\epsilon}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon}) \leq E_{\mathcal{P}+\epsilon}(\boldsymbol{x}^*_{\mathcal{P}})$. For the sake of notational convenience, we use $\boldsymbol{a}^* = \boldsymbol{x}^*_{\mathcal{P}}$ and $\boldsymbol{b}^* = \boldsymbol{x}^*_{\mathcal{P}+\epsilon}$. Now, from Corollary 1, we know that $\forall i \in \mathcal{V}, 0 \leq a_i^* \leq 1$. Therefore, we conclude that $\forall \boldsymbol{e}_{ij} \in \mathcal{E}, |a_i^* - a_j^*| \in [0, 1]$, and hence $\forall \epsilon \geq 0, \forall \boldsymbol{e}_{ij} \in \mathcal{E}, |a_i^* - a_j^*|^{p_{ij}+\epsilon} \leq |a_i^* - a_j^*|^{p_{ij}}$. As a consequence, we have the result that $E_{\mathcal{P}+\epsilon}(\boldsymbol{x}^*_{\mathcal{P}}) \leq E_{\mathcal{P}}(\boldsymbol{x}^*_{\mathcal{P}})$. The above results give us the relationship

$$E_{\mathcal{P}+\epsilon}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon}) \leq E_{\mathcal{P}+\epsilon}(\boldsymbol{x}^*_{\mathcal{P}}) \leq E_{\mathcal{P}}(\boldsymbol{x}^*_{\mathcal{P}}) \leq E_{\mathcal{P}}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon}). \tag{7}$$

Note that the term $E_{\mathcal{P}+\epsilon}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon}) - E_{\mathcal{P}}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon}) = \sum_{\boldsymbol{e}_{ij}\in\mathcal{E}} w_{ij}|b_i^* - b_j^*|^{p_{ij}}(1 - |b_i^* - b_j^*|^{\epsilon})$ is non-negative as per (7). In what follows, we attain an upper bound for this term, by finding the upper bound of $z^p - z^{p+\epsilon}, p \geq 1, \epsilon \geq 0$ under the constraint that $z \geq 0$. In particular, we find a $z^* > 0$ at which the derivative of $z^p - z^{p+\epsilon}$ vanishes.

$$\begin{aligned} &\frac{\partial(z^p - z^{p+\epsilon})}{\partial z}\Big|_{z=z^*} = pz^{*p-1} - (p+\epsilon)z^{*p+\epsilon-1} = 0 \\ &\implies z^* = \left(\frac{p}{p+\epsilon}\right)^{\frac{1}{\epsilon}} \in (0, 1). \end{aligned} \tag{8}$$

It can be checked that the second derivative at $z = z^*$ is negative, therefore establishing that $z^p - z^{p+\epsilon}$ attains its maximum value of $\frac{\epsilon}{p+\epsilon}\left(\frac{p}{p+\epsilon}\right)^{\frac{p}{\epsilon}} < \epsilon$ at $z = z^*$ (given by (8)). This implies that $0 \leq E_{\mathcal{P}+\epsilon}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon}) - E_{\mathcal{P}}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon}) = \sum_{\boldsymbol{e}_{ij}\in\mathcal{E}} w_{ij}[|b_i^* - b_j^*|^{p_{ij}} - |b_i^* - b_j^*|^{p+\epsilon}] \leq \epsilon \sum_{e_{ij}\in\mathcal{E}} w_{ij}$. It is important to observe that this is only a weak bound, but suffices for the purpose of our proof.

Therefore as $\epsilon \to 0$, we have that $\left(E_{\mathcal{P}+\epsilon}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon}) - E_{\mathcal{P}}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon})\right) \to 0$, and hence by (7), $\left(E_{\mathcal{P}}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon}) - E_{\mathcal{P}}(\boldsymbol{x}^*_{\mathcal{P}+\epsilon})\right) \to 0$. Since $E_{\mathcal{P}}(\boldsymbol{x})$ is continuous in $\boldsymbol{x}$, we conclude that as $\epsilon \to 0$, $\left(\boldsymbol{x}^*_{\mathcal{P}+\epsilon} - \boldsymbol{x}^*_{\mathcal{P}}\right) \to 0$, and hence the required result. ∎

## 4. Energy Minimization Via IRLS

In this section, we propose our iterative solver for the optimization problem of (2). We propose to use Iterative Reweighted Least Squares techniques that have been employed traditionally to minimize energy cost functions similar to $E_{\mathcal{P}}$ when $\forall \boldsymbol{e}_{ij} \in \mathcal{E}, p_{ij} = p \geq 1$. In what follows, we outline the steps of our algorithm and then discuss the convergence properties of the proposed algorithm.

Before we proceed, we formally define some terms that shall be used in our algorithm. Given the weights $\{w_{ij}\}$ for a graph $G$, we define a weights matrix $W \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, whose $(i, j)$ entry is given by $w_{ij}$ if $\boldsymbol{e}_{ij} \in \mathcal{E}$ and 0 otherwise. The matrix $W$ can be used to define a diagonal matrix $D \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, whose $i^{\text{th}}$ diagonal entry is given by $\sum_{j=1}^{|\mathcal{V}|} w_{ij}$. Finally, we define a Laplacian matrix $L(W) \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ as $L(W) = D - W$. Note that by construction of $L(W)$, we have that for any vector $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{V}|}, \boldsymbol{x}^\top L(W)\boldsymbol{x} = \sum_{\boldsymbol{e}_{ij}\in\mathcal{E}} w_{ij}(x_i - x_j)^2$. Therefore, if $\forall \boldsymbol{e}_{ij} \in \mathcal{E}, w_{ij} > 0$, then $L(W)$ is a positive-semi definite matrix with the only null vector being the vector with all entries equal to 1. We also decompose the matrix $L(W)$ into matrices such that

$$\boldsymbol{x}^\top L(W)\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_U & \boldsymbol{x}_M\end{bmatrix}\begin{bmatrix}L_U(W) & B(W)\\ B^\top(W) & L_M(W)\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_U\\ \boldsymbol{x}_M\end{bmatrix}. \tag{9}$$

**Algorithm 1** *(Numerical Scheme for Minimizing $E_{\mathcal{P}}(\boldsymbol{x})$).*

1. Set $n = 0$ and choose a value $\alpha > 0$.
2. Set the values of the potentials $\boldsymbol{x}_M$ at the marked nodes as $\boldsymbol{x}_i = 1$, if $i \in \mathcal{O}$ and $\boldsymbol{x}_i = 0$, if $i \in \mathcal{B}$. Initialize the values of the potentials at the unmarked nodes by setting them to 0, *i.e.* $\boldsymbol{x}_U^{(0)} = \mathbf{0}$.
3. Construct the weight matrix $W^{(n)}$ as

$$W_{ij}^{(n)} = \begin{cases} p_{ij} w_{ij} |x_i^{(n)} - x_j^{(n)}|^{p_{ij}-2} & \text{if } \boldsymbol{e}_{ij} \in \mathcal{E}, x_i^{(n)} \neq x_j^{(n)}, \\ \alpha^{p_{ij}-2} & \text{if } \boldsymbol{e}_{ij} \in \mathcal{E}, x_i = x_j, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

4. Estimate $\boldsymbol{x}_U^{(n+1)}$ as the minimizer of $\boldsymbol{x}^\top L(W^{(n)})\boldsymbol{x}$, that is given by $\boldsymbol{x}_U^{(n+1)} = -L_U^{-1}(W^{(n)})B(W^{(n)})\boldsymbol{x}_M$
5. If $|\boldsymbol{x}_U^{(n+1)} - \boldsymbol{x}_U^{(n)}| > \delta$, update $n = n+1$ and go to step 2.
6. Assign pixels $i$ with potential $x_i^{(n+1)} > 0.5$ to the object, and the remaining to background.

Given these definitions, we propose to minimize $E_{\mathcal{P}}(\boldsymbol{x})$ subject to the user specified constraints by using Algorithm 1. We shall now show that this iterative scheme descends the energy function, therefore outlining a proof of convergence. We separate our analysis into the cases (a) $\forall \boldsymbol{e}_{ij} \in \mathcal{E}, p_{ij} > 1$ and (b) $\forall \boldsymbol{e}_{ij} \in \mathcal{E}, p_{ij} \geq 1$. The reason for separating our analysis in this case is that for case (a), the function $E_{\mathcal{P}}(\boldsymbol{x})$ is differentiable and hence we can prove that our algorithm follows a framework similar to Newton descent. For the case (b), $E_{\mathcal{P}}(\boldsymbol{x})$ might not be differentiable. However, it is equivalent to the limit of the solutions given by a series of several algorithms that descend $E_{\mathcal{P}}(\boldsymbol{x})$.

**Case (a)**: $\boldsymbol{\forall e_{ij} \in \mathcal{E}, p_{ij} > 1}$: Note that the gradient of $E_{\mathcal{P}}(\boldsymbol{x})$ exists and is given as

$$\frac{\partial E_{\mathcal{P}}(\boldsymbol{x})}{\partial \boldsymbol{x}_U} = L_U(W)\boldsymbol{x}_U + B(W)\boldsymbol{x}_M. \tag{11}$$

Observe that the update vector at the $n^{\text{th}}$ step in Algorithm 1 is $\Delta \boldsymbol{x}_U^{(n)} = \boldsymbol{x}_U^{(n+1)} - \boldsymbol{x}_U^{(n)}$, which can be rewritten as

$$\begin{aligned} \Delta \boldsymbol{x}_U^{(n)} &= -L_U^{-1}(W^{(n)})B(W^{(n)})\boldsymbol{x}_M - \boldsymbol{x}_U^{(n)} \\ &= -L_U^{-1}(W^{(n)})[L_U(W^{(n)})\boldsymbol{x}_U^{(n)} + B(W^{(n)})\boldsymbol{x}_M] \\ &= -L_U^{-1}(W^{(n)})\frac{\partial E_{\mathcal{P}}(\boldsymbol{x}^{(n)})}{\partial \boldsymbol{x}_U}. \end{aligned} \tag{12}$$

Also, since $\forall \boldsymbol{e}_{ij} \in \mathcal{E}, W_{ij}^{(n)} > 0$, we can conclude that $L_U(W^{(n)})$ is a positive definite matrix [19]. Therefore, $\Delta \boldsymbol{x}_U^{(n)}$ is a descent direction at $\boldsymbol{x}_U = \boldsymbol{x}_U^{(n)}$, since $\Delta \boldsymbol{x}_U^{(n)} = -H\frac{\partial E_{\mathcal{P}}(\boldsymbol{x}^{(n)})}{\partial \boldsymbol{x}_U}$ is a descent direction for any positive definite $H$. Note that if $\forall \boldsymbol{e}_{ij} \in \mathcal{E}, p_{ij} = p > 1$, then $L_U(W)$ corresponds to the the Hessian of $E_{\mathcal{P}}(\boldsymbol{x})$. In this case, our method is equivalent to a Newton descent algorithm. If the norms vary across the edges, the matrix $L_U(W)$ is not equal to the Hessian of $E_{\mathcal{P}}(\boldsymbol{x})$, then the update is a descent direction different from the Newton descent direction.

**Case (b)**: $\boldsymbol{\forall e_{ij} \in \mathcal{E}, p_{ij} \geq 1}$: Recall from Theorem 2 that the minimizers of $E_{\mathcal{P}}(\boldsymbol{x})$ are right continuous with respect to the norms in $\mathcal{P}$. Now, consider an energy function $E_{\mathcal{P}}(\boldsymbol{x})$ such that for some edges $\boldsymbol{e}_{ij} \in \mathcal{E}$, the associated norm $p_{ij} = 1$. We propose to estimate the minimizer of $E_{\mathcal{P}+\epsilon}(\boldsymbol{x})$ using Algorithm 1, which as shown in case (a), provably converges to the minimizer $\boldsymbol{x}^*_{\mathcal{P}+\epsilon}$. Therefore, by choosing $\epsilon$ as small as possible, we can estimate $\boldsymbol{x}^*_{\mathcal{P}}$ with desired accuracy. As a result, the minimization can be seen as the limit of the results of several descent algorithms.

Note that in (10), we define the weight $W_{ij}^{(n)} = \alpha_{ij}^{p_{ij}-2}$ when $x_i = x_j$. This is in order to ensure that the inversion of $L_U(W^{(n)})$ in step 4 of Algorithm 1 is well-conditioned. In all our experiments, we used $\alpha = 10^{-6}$. Smaller values of $\alpha$ can result in the inversion still being ill-conditioned. Larger values of $\alpha$ do not accurately represent the condition $x_i = x_j$ and can hence result in slow convergence.

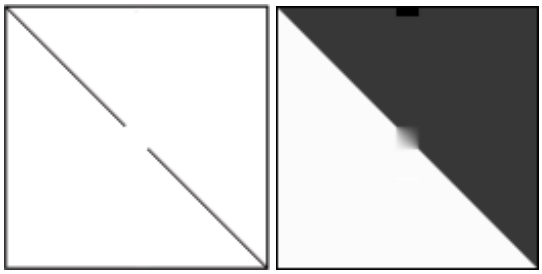
## 5. Results

In this section, we first present results on synthetic and real images to illustrate that an interpolated hybrid algorithm can be used to strike a desirable tradeoff between Graph Cuts and the Random Walker.

### 5.1. Testing on Synthetic Images

We test variants of our algorithm on the "diagonal line" image shown in Figure 1, to illustrate various positive/negative aspects of the different algorithms. We first examine the effect of an increasing $p$ value on the metrication (pixelation) error. When $p = 1$, the minimum cut is degenerate in this synthetic image. Traditional maximum flow algorithms typically find a "squared off" cut. However, our iterative method can instead choose the desired diagonal cut, where thresholding of the potential field shown in Figure 1, permits any of the degenerate boundaries.

If we break the degeneracy by modifying the image to include a single speck in the ambiguous region, the "squared off" metrication effect is seen once again. These results are exhibited in Figure 2. The Random Walker ($p = 2$) still finds the smoother diagonal cut. We increase the value of $p$ to observe the transition between the blocky $p = 1$ case and the smooth $p = 2$ case. We see that the metrication error disappears for values of $p \geq 1.5$. When $p$ is pushed above $p = 2$, the solution begins to resemble the distance-based boundary for the $p = \infty$ case, as predicted in [20].



(a) "diagonal line" image (b) Potentials obtained for $p$=1

Figure 1. Unlike traditional max-flow solutions for the $p = 1$ case, our method gives continuous values which may be thresholded at 0.5 to produce the desired smooth, diagonal cut.

Now, we examine the effect of $p$ on the proximity bias known to occur for Random Walker ($p = 2$). In this case, it is known previously that for Graph Cuts ($p = 1$), there should be no dependency on seed location within either the upper or lower region. In Figure 3, we examine how intermediate values of $p$ effect this bias. We see that the "bulge" problem of the Random Walker is avoided for $p \leq 1.5$,.

Finally, we probe the shrinking bias for various $p$ values. Given a single seed and a weak boundary, Graph Cuts produces a segmentation that minimally surrounds the seed while the Random Walker algorithm will avoid this problem. We need to address the question: At what value of $p$ does the shrinking bias disappear? Figure 4 shows the segmentations obtained for various values of $p$. We see that the shrinking bias is removed quickly as $p$ increases, and in this case, is gone after $p \geq 1.2$. The shrinking bias is further illustrated in Figure 5, on an image with concentric circles. Since Graph Cuts has a bias towards segmentation boundaries of shorter length, it places the boundary at the innermost circle. As we increase the value of $p$, the segmentation boundary moves towards the outer circle. Interestingly, even when we increase the value of $p \geq 3$, we are unable to produce the segmentation boundary beyond the midpoint circle since the solution approaches the shortest paths solution, which depends only on the number of circles necessary to cross over to the foreground from the background.

In conclusion, these tests on synthetic examples motivate the fact that that the problems of Graph Cuts (shrinking bias and metrication) and Random Walker (proximity bias) can both be avoided by using an intermediate value of $p$, such as $1.25 \leq p \leq 1.5$, which produces a hybrid algorithm that does not suffer from the problems of either algorithm.

### 5.2. Testing on Real Images

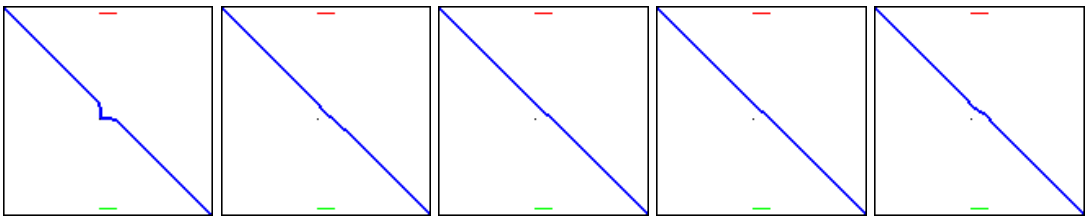
In this section we applied algorithms with varying $p$ to the set of images in the Microsoft "GrabCut" database used in [17] .The database contains ground truth segmentations for 50 color images corresponding to indoor as well as outdoor scenes. We use a 4-connected lattice and use the Euclidean distances of the RGB values of these color images to set image weights. No unary (data) terms were used. In our experiments, we use two different seeding strategies. In the first case, we use the seeds provided in the database itself, obtained by eroding the ground truth segmentation. Since such scribbles are favorable for distance based segmentation schemes, we also consider a new set of scribbles. The scribbles for the background are taken as the rectangular bounding boxes provided by the database. We dilate the object's skeleton and use it as a scribble for the object.

We estimate the segmentation using different values of $p$, where we use exactly the same graph/weights for all values of $p$. Note that the $p = 1$ case was produced using a Graph Cuts implementation rather than our IRLS algorithm. We quantify the error in the results using four different standard segmentation measures used in [25] , namely Boundary Error (BE), Rand Index (RI), Global Consistency Error (GCE), and Variation of Information (VoI). Good segmentation quality is marked by low BE, high RI, low GCE and low VoI. Results of this test of segmentation quality are given in Tables 1 and 2. From these segmentation measures, we produce a rank for each $p$ value with respect to its performance for each metric, and compute an average rank (Avg. rank) by taking the average of the ranks for different metrics.

Figures 6 and 7 show typical results obtained in this analysis. The seeds are marked in red and green for the foreground and background respectively. As expected, we see that the first seeding scenario (native to the database), results in segmentation errors that improve as we increase the value of $p$. This is due to the fact that proximity bias tends to increase with the value of $p$. In the second seeding scheme, we see that the results are better for lower values of $p$ because the seed locations are extremely poor indicators of the desired segmentation boundary. However, we notice that across both seeding schemes, the quality score of $p = 1.5$ indicates a desirable tradeoff between Graph Cuts ($p = 1$) and Random Walker ($p = 2$). One can always choose Graph Cuts or the Random Walker if one is aware of the seeding schemes that are going to be employed in a particular application. However, from the point of a general interactive segmentation algorithm, the $p$-brush algorithm with $p = 1.5$ has the best rank on average across seeding strategies.

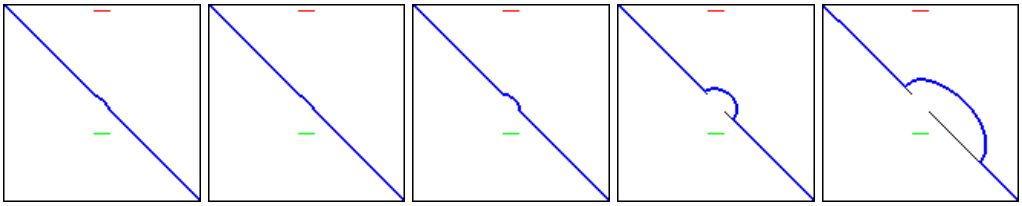
## 6. Comments

In this paper, we have described a family of segmentation algorithms dependent on a parameter $p \geq 1$, that has Graph Cuts and Random Walker as special cases ($p = 1$ and $p = 2$, respectively). We provide a solver for arbitrary values of $p$ and show that an intermediate value between $1.25 \leq p \leq 1.5$ provides a segmentation algorithm that avoids the known problems of Graph Cuts and Random Walker. The solver is provably convergent only for $1 < p < 3$ [15], but convergence issues for $p \geq 3$ can be resolved by taking adaptive step sizes as opposed to a fixed step size (1 in our case).



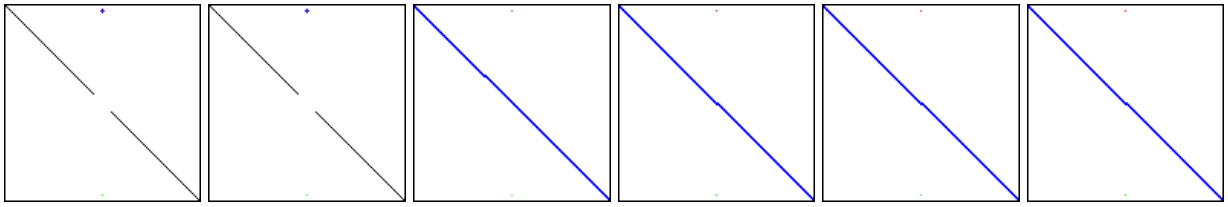
(a) $p = 1.05$ boundary (b) $p = 1.1$ potentials (c) $p = 1.5$ potentials (d) $p = 2.0$ potentials (e) $p = 3.0$ potentials

Figure 2. In this experiment, we vary $p$ between 1 and 3 in order to determine when the metrication problem disappears. The foreground and background seeds are indicated in green and red, respectively. The segmentation boundary estimated by the algorithm is shown in blue.



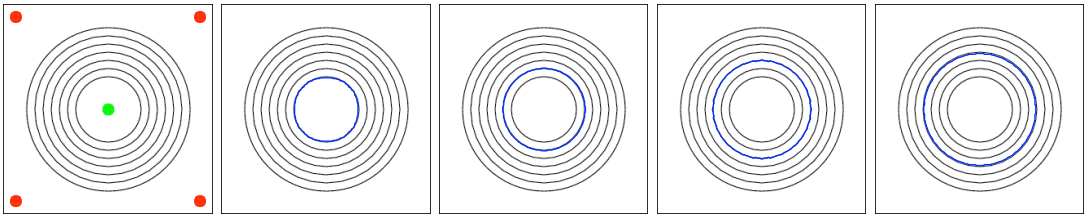
(a) $p = 1.05$ boundary (b) $p = 1.25$ boundary (c) $p = 1.5$ boundary (d) $p = 2.0$ boundary (e) $p = 3.0$ boundary

Figure 3. In this experiment, we vary $p$ between 1 and 3 to study of the effects of $p$ on proximity bias.The foreground and background seeds are indicated in green and red, respectively. The estimated segmentation boundary is shown in blue.



(a) $p = 1.0$ boundary (b) $p = 1.1$ boundary (c) $p = 1.2$ boundary (d) $p = 1.5$ boundary (e) $p = 1.75$ boundary (f) $p = 2.0$ boundary

Figure 4. In this experiment, we vary $p$ between 1 and 2 in order to determine when the shrinking bias disappears. The foreground and background seeds are indicated in green and red, respectively. The segmentation boundary estimated by the algorithm is shown in blue.



(a) Image + seeds (b) $p = 1$ boundary (c) $p = 1.5$ boundary (d) $p = 2$ boundary (e) $p = 2.9$ boundary

Figure 5. In this experiment, we further probe the shrinking bias. The foreground and background seeds are shown in green and red, respectively. The segmentation boundary is overlaid in blue. Since the shrinking bias is largest for smaller values of $p$, the boundary is placed at the inner most circle for $p = 1$. As we increase $p$, the shrinking bias reduces and the segmentation boundary moves outwards.

Note that we did not compare our framework with [17, 14, 24, 4], since these methods have an an elaborate pipeline for segmentation, with an intermediate step which uses schemes such as Graph Cuts, Random Walker or shortest paths for MAP-MRF. Hence, our analysis and algorithm is relevant to improving the properties/performance of the MAP-MRF submodule in such intermediate steps, and should not be unfairly compared to the entire algorithms. Specifically, [17] employs Graph Cuts, while [14, 24] use a quadratic ($p = 2$) optimization with the Laplacian matrix which may be interpreted as Random Walker with special weights. Similarly, [4] employ shortest paths for MAP-MRF, which is equivalent to $p \rightarrow \infty$ in our framework.

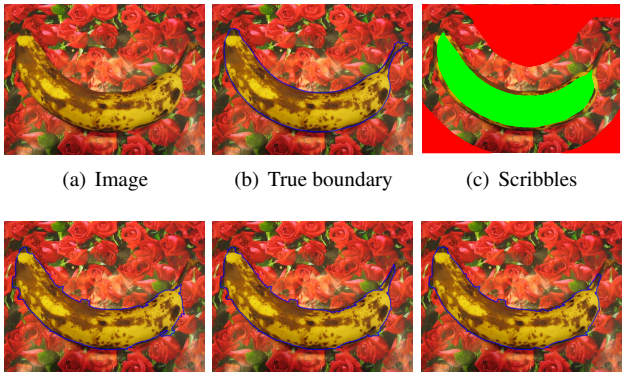
Also, despite the recent popularity of higher order MRFs [16, 12, 13], we feel that pairwise MRFs are still used ubiquitously in the computer vision literature and any improvement to these models can have a broad impact. Recent progress in Graph Cut based approaches for higher order MRFs [16, 12, 13] has dealt with representing a class of higher order MRFs by an equivalent construction of pairwise MRFs. Therefore, our analysis could be applied to this equivalent pairwise reduction of higher order MRF models.

| $p$ | BE | RI | GCE | VoI | Avg. rank |
|---|---|---|---|---|---|
| 1. | 3.276 | 0.970 | 0.028 | 0.196 | 5 |
| 1.25 | 3.241 | 0.971 | 0.028 | 0.193 | 4 |
| 1.5 | 3.214 | 0.972 | 0.027 | 0.189 | 3 |
| 1.75 | 3.206 | 0.972 | 0.027 | 0.187 | 2 |
| 2 | 3.206 | 0.972 | 0.026 | 0.185 | 1 |

Table 1. Mean errors using the eroded ground truth as scribbles

| $p$ | BE | RI | GCE | VoI | Avg. rank |
|---|---|---|---|---|---|
| 1 | 8.713 | 0.916 | 0.393 | 0.068 | 1.25 |
| 1.25 | 11.184 | 0.908 | 0.408 | 0.067 | 2.5 |
| 1.5 | 11.487 | 0.907 | 0.406 | 0.067 | 2.25 |
| 1.75 | 12.292 | 0.904 | 0.417 | 0.069 | 4 |
| 2 | 12.727 | 0.900 | 0.432 | 0.071 | 5 |

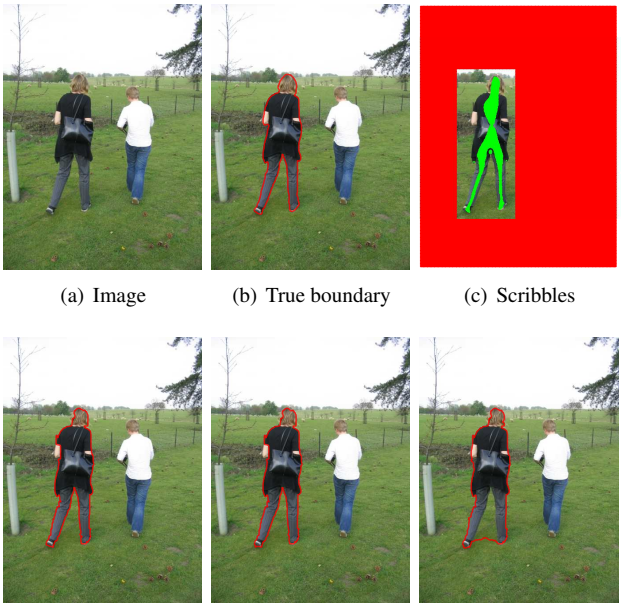
Table 2. Mean errors using the rectangle/skeleton as scribbles



(a) Image (b) True boundary (c) Scribbles

(d) $p = 1$ boundary (e) $p = 1.5$ boundary (f) $p = 2$ boundary

Figure 6. Example of testing using the native GrabCut seeds. The segmentation boundaries are marked in blue. Graph Cuts ($p = 1$) shows shrinking bias and misses the tip of the banana. This effect is not present in the cases of $p = 1.5$ and $p = 2$ (Random Walker).



(a) Image (b) True boundary (c) Scribbles

(d) $p = 1$ boundary (e) $p = 1.5$ boundary (f) $p = 2$ boundary

Figure 7. Example of testing using the second seeding strategy. The segmentation boundaries are marked in red. The Random Walker ($p = 2$) is sensitive to the scribble near the woman's leg. This is not so for the cases of $p = 1$ (Graph Cuts) and $p = 1.5$.

## References


[1] C. V. Alvino, G. B. Unal, G. G. Slabaugh, B. Peny, and T. Fang. Efficient segmentation based on Eikonal and diffusion equations. *Int. J. Comput. Math*, 84(9):1309–1324, 2007.

[2] A. X. Falcao, R. D. A. Lotufo, and G. Araujo. The image foresting transformation. *IEEE PAMI*, 26(1):19–29, 2004.

[3] B. Appleton and H. Talbot. Globally optimal surfaces by continuous maximal flows. *IEEE PAMI*, 28(1):106–118, 2006.

[4] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV*, 2007.

[5] Y. Boykov and M. Jolly. Interactive Graph Cuts for optimal boundary & region segmentation of objects in N-D images. In *ICCV*, pages 105–112, 2001.

[6] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via Graph Cuts. In *ICCV*, 2003.

[7] R. Coifman and M. Wickerhauser. Entropy-based algorithms for best bases selection. *IEEE Transactions on Information Theory*, 38(2):713–718, 1992.

[8] A. Criminisi, T. Sharp, and A. Blake. Geos: Geodesic image segmentation. In *ECCV (1)*, pages 99–112, 2008.

[9] B. P. D. Greig and A. Seheult. Exact maximum *a posteriori* estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.

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

[11] L. Grady and A. Sinop. Fast approximate Random Walker segmentation using eigenvector precomputation. In *CVPR*, 2008.

[12] P. Kohli, M. P. Kumar, and P. H. S. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[13] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[14] A. Levin, D. Lischinski, and Y. Weiss. A Closed Form Solution to Natural Image Matting. In *CVPR*, pages 61–68, 2006.

[15] M. R. Osborne Finite algorithms in optimization and data analysis. *Wiley Series in Probability and Mathematical Statistics*, 1985.

[16] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *CVPR*, pages 860–867, 2005.

[17] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

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

[19] D. Singaraju, L. Grady, and R. Vidal. Interactive image segmentation via minimization of quadratic energies on directed graphs. In *CVPR*, 2008.

[20] A. Sinop and L. Grady. A seeded image segmentation framework unifying Graph Cuts and Random Walker which yields a new algorithm. In *ICCV*, 2007.

[21] M. Tappen, L. Ce, E. Adelson, and W. Freeman. Learning Gaussian conditional random fields for low-level vision. In *CVPR*, 2007.

[22] M. Unger, T. Pock, D. Cremers, and H. Bischof. TVSeg - interactive total variation based image segmentation. In *BMVC*, 2008.

[23] S. Vicente, V. Kolmogorov, and C. Rother. Graph Cut based image segmentation with connectivity priors. In *CVPR*, 2008.

[24] J. Wang and M. F. Cohen. Optimized color sampling for robust matting. In *CVPR*, 2007.

[25] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *CVIU*, 110(2):212–225, 2008.