A Quadratic Programming Approach to Image Labeling

Zhouyu Fu[1, 2] and Antonio Robles-Kelly[1, 2]

[1]RSISE, Bldg. 115, Australian National University, Canberra ACT 0200, Australia

[2]National ICT Australia (NICTA)*, Locked Bag 8001, Canberra ACT 2601, Australia

Abstract

Image labeling tasks are usually formulated within the framework of discrete Markov Random Fields where the optimal labels are recovered by extremising a discrete energy function. In this paper, we present an alternative continuous relaxation approach to image labeling which makes use of a quadratic cost function over the class labels. The cost function to minimise is convex and its discrete version is equivalent up to a constant additive factor to the target function used in discrete Markov Random Field approaches. Moreover, its corresponding Hessian matrix is given by the graph Laplacian of the adjacency matrix. Therefore, the optimisation of the cost function is governed by the pairwise interactions between pixels in the local neighbourhood. This leads to a sparse Hessian matrix for which the global minimum of the continuous relaxation problem can be efficiently found by solving a system of linear equations using Cholesky factorisation. We elaborate on the links between the method and other techniques elsewhere in the literature and provide results on synthetic and real-world imagery. We also provide comparison to competing approaches.

1 Introduction

Many classical problems in computer vision, such as image segmentation and denoising can be recast as image labelling ones. The purpose of image labeling is to assign each image element

*NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

(pixels, patches, features, etc.) to one of the classes under consideration so as to optimise a target function. This is based upon an optimality criterion which takes into account the affinity of the image elements to the assigned class and the consistency of the class membership with respect tot he neighbouring elements.

Approaches such as the relaxation labeling [10] try to solve the problem by enforcing hard constraints on the consistency term of the target function, which is governed by the pairwise relationships between pixels. They do this by searching for a labeling scheme so as to minimize a cost function subject to constraints. A related approach is that based upon Markov Random Fields (MRFs) [8]. MRFs are a statistical framework underlying many image processing and low level vision applications. The MRF model is a powerful one, which encodes statistical dependencies between neighbouring image elements by viewing them as nodes in a graphical model. The weight of each node is given by the cost of assignment based on unitary relationships. The edges between adjacent nodes in the graph represent the costs of concurrent assignments of neighbouring pixels.

The extremisation of the discrete energy function arising from MRFs is often achieved through continuous relaxation schemes which aim at transforming the original combinatorial optimisation problem into a continuous one that can be solved by mathematical programming techniques such as Iterative Conditional Means (ICM), gradient based optimisation and simulated annealing [15]. These methods have their respective drawbacks. ICM and gradient based optimisation, although being efficient, are known to be sensitive to initialisation and are easily trapped in local minima. Simulated annealing, on the other hand, has a very high computational complexity and convergence to global optimality is only guarrantteed in theory by taking infinitesimal annealing steps [**?**]. This limits its application to image labeling problems with hundreds of thousands of pixel-labels to recover.

Current methods for inference over MRFs can be roughly divided into two categories based on the representation they employ for the label-sets. The first category comprises those approaches which represent labels in their original discrete form and tries to recover them directly. Maybe the two most popular representatives of this group are graph cuts [1, 3, 13] and belief propagation [7]. The second category of methods encodes labels as continuous variables and minimises a continuous cost function for which the final labeling scheme is obtained by discretising the solution of the continuous problem. A number of computational methods have been used to formulate the problem in a continuous domain. These mathematical programming methods include Quadratic Programming (QP) [17], Semi-definite Programming (SDP) [12, 11, 21], Second Order Cone Pro-

gramming (SOCP) [14] and Spectral Relaxation [5].

Both, the discrete and the continuous relaxation methods have their own merits and limitations for solving the MRF-based labelling problems. For MRFs with binary labels, graph cuts are very efficient and can achieve global minimum of the cost function [3] in so far that the term in the energy function containing the pairwise potential satisfies submodular conditions [13]. Graph cuts based on quadratic pseudo boolean optimisation [19] can handle energy functions with non-submodular pairwise terms. Unfortunately, quadratic pseudo boolean optimisation is less powerful than submodular graph cuts in terms of efficiency and there is no guarantee for global optimality.

In the other hand, continuous relaxation methods can encode arbitrary pairwise costs. Their main drawback relates to their computational cost, which makes them much slower than their discrete counterparts. Moreover, the approximation from the recovered continuous solution to the discrete labeling is quite obvious. This approximation is indispensable for continuous methods since, otherwise, the integer programming problem, which is NP hard, would have to be solved. A less obvious approximation is in the minimisation of the continuous function. Despite the promise of low computational cost, as we discuss later, the continuous function yielded by directly relaxing the discrete variables of the function of pairwise MRFs is not necessarily a convex one. Hence, a global minimum of the cost function is not guaranteed.

The remainder of this paper is as follows. Section 2 briefly outlines the contribution of this paper. Section 3 provides the background on MRF models for image labeling used throughout the paper. Section 4 details our approach to the labeling problem. Since our method hinges in the use of continuous relaxation, we elaborate on the discrete label recovery and provide the algorithm description in Section 5. In Section 6, we provide further discussion on the approach presented here, its relation to random walks, diffusion processes and the choice of optimisation procedure. Experimental results are provided in Section 7. Conclusions are given in Section 8.

2 Contribution

Here we aim at overcoming these drawbacks, i.e. namely the complexity and the non-convex nature of the relaxed continuous cost function. To this end, we propose an alternative approach to solving the MRF inference problem which hinges in the minimisation of a continuous cost function making use of a graph regularisation scheme over a manifold [22]. By ignoring the constraints on the label field, we note the equivalence, up to an additive factor, of the graph regularisation

cost function presented here and that corresponding to the MRF target function when discrete labels are considered. Constraints can then be absorbed into the cost function so as to develop a constrained graph regularisation framework for image labeling. The optimal labels are, hence, given by minimising the cost function and discretising the obtained solutions into binary values. Further, the cost function is convex and, thus, it is not susceptible to local minima. We show that our approach can be reduced to a linear system of equations, where the Hessian is a sparse positive-semidefinte matrix determined by the pairwise costs between neighbouring pixels. Hence, we can make use of a direct algorithm for solving the linear system efficiently making use of Cholesky factorisation [6].

The method presented here is quite general in nature. Indeed, our approach can be applied to binary and multiclass image labeling problems alike. Moreover, the algorithm is not restricted to colour-pixel values. The image elements to be labelled can either be pixels or patches, both of which are represented by nodes and connections between them in the graph. Further, due to the probabilistic nature of the obtained label vectors, we can naturally generalise the framework to handle labeling problems with continuous label values, such as those found in image matting problems.

3 MRF for Image Labeling

As mentioned earlier, Markov Random Field (MRF) models are a probabilistic framework for solving labeling problems over structured fields. In this section, we provide the notation and basic theoretical foundations used throughout the paper. To commence, we require some formalism. Let $G(\mathcal{V}, \mathcal{E})$ denote a graph with node-set $\mathcal{V} = \{V_1, \ldots, V_N\}$ and edge-set $\mathcal{E} = \{E_{i,j} | V_i \sim V_j \text{ and } V_i, V_j \in \mathcal{V}\}$ where $\sim$ denotes the neighbourhood relationship. Each $V_i \in \mathcal{V}$ is associated with a label X_i which takes a set of discrete values in $\{1, \ldots, K\}$, where K is the number of label classes. The joint probability distribution of the label field X represented by the MRF is given by

$$p(X|\theta) = \frac{1}{Z(\theta)} P(X) = \frac{1}{Z(\theta)} \prod_{i,j \in \mathcal{E}} \psi_{i,j}(X_i, X_j|\theta_{i,j}) \prod_{i \in \mathcal{V}} \phi_i(X_i|\theta_i) \tag{1}$$

$$Z(\theta) = \int_X P(X|\theta) dX = \int_X \prod_{i,j \in \mathcal{E}} \psi_{i,j}(X_i, X_j|\theta_{i,j}) \prod_{i \in \mathcal{V}} \phi_i(X_i|\theta_i) dX$$

where $\phi_i(X_i|\theta_i)$ and $\psi_{i,j}(X_i, X_j|\theta_{i,j})$ are unitary and binary potential functions which determine the likelihood of the assignment of nodes in the graph to the label classes, and $\theta = \{\theta_i, \theta_{i,j}\}$ is

the set of parameters that specifies the unary and binary potential functions. $Z(\theta)$ is the partition function, i.e. the normalisation factor to make $p(X)$ a proper probability distribution function with $\int p(X)dX = 1$. $Z(\theta)$ is a function of the parameter-set θ and indpendent of X, as it is obtained by marginalising $P(X)$, the product of potential functions, with respect to X. The inference of the above MRF model can therefore be recast as an maximum a posteriori (MAP) estimation problem so as to maximise $P(X)$ over all possible assignments of $X = \{X_i | i = \{1, \ldots, K\}$.

For the sake of convenience, the following negative log-likelihood function is usually considered instead of the joint distribution function in Equation 1.

$$q(X) = -\log P(X) = \sum_{i=1}^{N} c_i(X_i) + \sum_{i \sim j} v_{i,j}(X_i, X_j) \tag{2}$$

where $c_i(X_i) = -\log \phi_i X_i|\theta_i$, $v_{i,j}(X_i, X_j) = -\log \psi_{i,j}(X_i, X_j|\theta_{i,j})$ are the negative logarithm of the corresponding potential functions. To solve the above problem, we introduce label vector $[x_{i,1}, \ldots, x_{i,K}]$ to represent the label of node i. Specifically, a $0-1$ encoding scheme is adopted where $x_{i,j} = 1$ if and only if the ith node is assigned label j and $x_{i,j} = 0$ otherwise. Hereafter, we will use the shorthand X_i to represent the label vector introduced. This should not be confused with the discrete label values X_i used in Equation 1. Their distinction should be clear from the context.

With the above ingredients, Equation 2 can be rewritten as

$$\begin{aligned} q(X) = -\log P(X) &= \sum_{i=1}^{N} \sum_{a=1}^{K} c_i(a) x_{i,a} + \sum_{i \sim j} \sum_{a=1}^{K} \sum_{b=1}^{K} v_{i,j}(a,b) x_{i,a} x_{j,b} \\ x_{i,a} \in \{0,1\} \quad &\text{and} \quad \sum_{a} x_{i,a} = 1 \end{aligned} \tag{3}$$

where i, j are node indices, and a, b are label indices. $c_i(a) = -\log \phi_i(a)$ and $v_{i,j}(a,b) = -\log \psi_{i,j}(a,b)$ are the same as those defined in Equation 2. The constraints guarranttee that one and only one of $x_{i,a}$ in label vector X_i can have value of 1 for each i. It can be easily seen that the problem defined in Equation 3 is equivalent to the problem in Equation 2. Hence minimising the above cost function is also equivalent to solving the original MAP-MRF inference problem, as defined in Equation 1. However, the problem is NP hard due to the binary constraints over the label variables $x_{i,a}$. To make the optimisation problem tractable, we can relax the discrete variables $x_{i,a}$ in the above equation by replacing the binary constraints $x_{i,a} \in \{0,1\}$ with the bound constraints $x_{i,a} \in [0,1]$. The cost function of this relaxed optimisation problem is in quadratic form and, hence, it is natural to apply quadratic programming techniques to solve it [17]. However, the

Hessian of Equation 3 is determined by the coefficients of the second order term $w_{i,j}(a,b)$ and is in general not necessarily positive semidefinite. Consequently, the cost function is not guaranteed to be convex and there is no optimality guarantee even for the relaxed problem.

A number of techniques have been proposed to deal with the discrete labeling problem and convert the MRF cost function into more tractable usually convex formulations. Along these lines, some examples of this are semidefinite programming (SDP) [12, 11, 21], second-order cone programming (SOCP) [14], and spectral relaxation [5]. Furthermore, these methods are computationally quite expensive and their applications are often restricted to small X-sizes. This limits their applicability to image labeling problems, where even moderate image sizes yield hundreds of thousands of pixel labels to resolve. To remedy this, in practice, a pre-segmentation step is effected so as to reduce the size of the problem to be handled by the numerical programming algorithm. [12, 11].

The pre-segmentation step as described above implies a cost in segmentation quality, as there is no way to recover from those errors made in the pre-segmentation stage. An alternative to preprocessing is that of a simplified cost function. Along these lines, the random walker segmentation algorithm [9] makes use of a simplified data term and a convex quadratic term for purposes of enforcing smoothness. As we show later, the random walker algorithm is a special case of the approach proposed in this paper.

4 Labeling via Graph Regularisation

4.1 Problem Formulation

As an alternative to a continuous relaxation corresponding to the cost function in Equation 3, here we present an alternative functional which is closely related to the MRF formulation. Note that the first term on the right-hand-side of Equation 3 measures the compatibility between the label and the unary node potential. The second term relates to the compatibility between labels of neighbouring nodes given by the pairwise potential $w_{i,j}(a,b)$. This term can be viewed as a correlation one between pixel-label values.

By thinking of correlation as a measure of similarity and viewing it an inverse distance, we can transform the maximisation problem at hand into a minimisation one. To this end, we employ the L_2 norm. This is a natural choice which leads to an efficient solution for the label values. The

corresponding cost function is hence defined as follows

$$\min f(X) = \sum_{i=1}^{N}\sum_{a=1}^{K}\gamma_a(X_i) + \sum_{i\sim j}\sum_{a=1}^{K}\sum_{b=1}^{K} w_{i,j}(a,b)(x_{i,a} - x_{j,b})^2 \tag{4}$$

$$\gamma_a(X_i) = \sum_{k\neq a} c_i(k)(x_{i,k} - 1)^2 + c_i(a)x_{i,a}^2 \tag{5}$$

$$w_{i,j}(a,b) = \sum_{a'\neq a}\sum_{b'\neq b} v_{i,j}(a',b') \tag{6}$$

$$x_{i,a} \in [0,1] \quad \text{and} \quad \sum_{a} x_{i,a} = 1$$

The above reformulation of the cost function has a number of advantages. Firstly, it is closely related to the MRF model in terms of its physical meaning. Like the MRF, our cost function accommodates two complementary terms, i.e. a term which measures the compatibility between the pixel data and its assigned label and a smoothness term which enforces consistency between labels for neighbouring nodes. The main difference between the cost function above and that in Equation 3 is the replacement of the inner product with a squared distance. We can further show that the cost function in Equation 4 shares the same optimal binary value solution as that in Equation 3. The proof is given in Appendix A.

As a consequence, the optimal solution obtained in the compact domain $[0, 1]$ making use of the reformulated MRF energy function, as given in Equation 4, is a binary vector. Here, we note that, the largest continuous label-value has a clear margin over those corresponding to other classes, we use a majority voting scheme by assigning the largest value in each X_i to 1 and the remaining to 0. If there is ambiguity regarding the labeling, i.e. the rank-difference between label-values is small, we adopt a strategy based on the structure of the label-field as a postprocessing step. We will elaborate on this label assignment procedure later in the paper. For now, we continue our analysis of the alternative cost function above.

It is straightforward to note that the reformulated cost function in Equation 4 is a convex quadratic program with a sparse Hessian matrix. The convexity immediately follows from triangular inequality. This is as a result of the squared distance term on the right-hand-side of the equation. The consequences of the cost function convexity are twofold. Firstly, we can always attain globally optimal solutions for the relaxed problem on the continuous label variables. Secondly, we can drop the inequality constraints regarding the non-negativity of the variable X_i as the solution to the unconstrained function naturally satisfies the non-negativity constraints. To see

this, we note that the solution of Equation 4, as determined by the data term alone is given by

$$\hat{X}_i = \frac{1}{Z}[C_i - c_i(1), \ldots, C_i - c_i(K)] \tag{7}$$

where $C_i = \sum_{a=1}^{K} c_i(a)$ and Z is the normalisation factor that makes X_i a probability measure over the label field. Thus, the recovered X_i is a non-negative vector with unit sum.

As we will discuss in Section 6, the pairwise weight $w_{i,j}(a,b)$ is also non-negative. This binary term is equivalent to a diffusion process over the graph, which propagates label values between neighbouring pixels so as to minimise the overall energy function. Nonetheless the sum-to-one constraints pertaining the binary term have to be taken into account, these can be absorbed into the cost function. Lastly, since an image is usually represented by a lattice-like graph, where each pixel is only connected to its neighbours on a grid, the corresponding graphical model leads to a sparse Hessian matrix. Thus, the problem can be reduced to that of solving a sparse linear system of equations with positive semidefinite Hessian. As we show later in the paper, a sparse variant of the Cholesky transform [6] can be used to solve this linear system efficiently.

4.2 Solution

To extremise the cost function in Equation 4, we commence by considering the sum-to-one constraints on the variable X_i. We note that these can be absorbed into the cost function turning the task in hand into an unconstrained quadratic program. To simplify the notation, we first rewrite the cost function as follows

$$\min f(X) = \sum_{i=1} ||X_i - \hat{X}_i||_C^2 + \sum_{i \sim j} \sum_a \sum_b w_{i,j}(a,b)(x_{i,a} - x_{j,b})^2 \tag{8}$$

where C is a diagonal matrix whose diagonal elements are the variables C_i and $\hat{X}_i$ are the as given in Equation 7 and $||d||_C^2 = d^T C d$ is a quadratic term.

Let $\mathrm{vec}(\hat{X})$ and $\mathrm{vec}(X)$ denote the vectorial forms of the matrices C and X of order $N \times K$. These vectors are conformed by the column-wise concatenation over the matrices. With these ingredients, we can rewrite Equation 4 in compact form as follows

$$\begin{aligned} \min f(\mathrm{vec}(X)) \quad &= \quad \frac{1}{2}||\mathrm{vec}(\hat{X}) - \mathrm{vec}(X)||_{\tilde{C}}^2 + \frac{1}{2}\mathrm{vec}(X)^T L \mathrm{vec}(X) \\ s.t. \quad &\quad A^T \mathrm{vec}(X) = b \end{aligned} \tag{9}$$

where $\tilde{C} = I_{K,K} \otimes C$, $L = D - \tilde{W}$ is the Laplacian of the graph and $A = 1_{K,1} \otimes I_N$ and $b = 1_{N,1}$, where $1_{K,1}$ is a vector of all 1's and I_N the identity matrix whose dimensionalities are denoted by

their respective subscripts. In the equation above, $\tilde{W}$ denotes the $NK \times NK$ adjacency matrix collapsed from the 4-tensor field W, whose $[i + N(a-1), j + N(b-1)]$th entry is given by $W(i,a,j,b)$ and $D = diag(\tilde{W}1_{N,1})$ is the degree matrix of $\tilde{W}$. Recall that D is a diagonal matrix whose elements are determined by the row sum of $\tilde{W}$.

At this point, it is worth noting that L is positive semidefinite [4] and so is the Hessian of the quadratic cost function. Moreover, the matrix constraint in the above equation permits subsuming the sum-to-one constraints for each X_i and absorbing the sum-to-one constraint into the cost function. To do this, we define a set of variables $\overline{X} = [X_{.,1}, \ldots, X_{.,K-1}]$ making use of the first $K-1$ columns of the label matrix X and $\mathrm{vec}(\overline{X})$, i.e. the vectorial form of $\overline{X}$. Since the the Kth column of X is given by

$$\begin{aligned} X_{,K} &= 1_{N,1} - \sum_j X_{.,j} = 1_{N,1} - E\mathrm{vec}(\overline{X}) \\ E &= 1_{1,K-1} \otimes I_{N,N} \end{aligned} \tag{10}$$

we can represent $\mathrm{vec}(X)$ in terms of $\mathrm{vec}(\overline{X})$ in the following manner

$$\begin{aligned} \mathrm{vec}(X) &= \begin{bmatrix} \mathrm{vec}(\overline{X}) \\ 1_{N,1} - E\mathrm{vec}(\overline{X}) \end{bmatrix} = u - F\mathrm{vec}(\overline{X}) \\ &= \begin{bmatrix} O_{N(K-1),1} \\ 1_{N,1} \end{bmatrix} - \begin{bmatrix} I_{N(K-1)} \\ E \end{bmatrix} \mathrm{vec}(\overline{X}) \end{aligned} \tag{11}$$

By substituting Equation 11 into Equation 9 we obtain the unconstrained cost function as defined with respect to $\mathrm{vec}(\overline{X})$, which is given by

$$\begin{aligned} \min f(\mathrm{vec}(\overline{X})) &= ||\mathrm{vec}(\hat{X}) - u + F\mathrm{vec}(\overline{X})||^2_{\tilde{C}} + (u - F\mathrm{vec}(\overline{X}))^T L(u - F\mathrm{vec}(\overline{X})) \\ &= \mathrm{vec}(\overline{X})^T F^T (L + \tilde{C}) F\mathrm{vec}(\overline{X}) + 2(\mathrm{vec}(\hat{X})^T \tilde{C} - u^T \tilde{C} - u^T L) F\mathrm{vec}(\overline{X}) \\ &+ ||\mathrm{vec}(\hat{X}) - u||^2_{\tilde{C}} + u^T L u \end{aligned} \tag{12}$$

Thus, we can extremise the cost function directly by taking the gradient of Equation 12 with respect to $\mathrm{vec}(\overline{X})$ and equating it to zero. This yields

$$\begin{aligned} H\mathrm{vec}(\overline{X}) &= d \\ H &= F^T (L + \tilde{C}) F \\ d &= u^T \tilde{C} F + u^T L F - \mathrm{vec}(\hat{X})^T \tilde{C} F \end{aligned} \tag{13}$$

where F and I_{NK} are sparse matrices. Although the node-set can be very large, this is expected, since, each node in the graph is only connected to its local neighbourhood. Therefore the adjacency matrix W/$\tilde{W}$ is sparse and so is the graph Laplacian L. The sparsity of the graph Laplacian also implies a sparse Hessian matrix whose number of elements is determined by $O(K|W|)$, where $|W| = N|m|$ is the total number of edge links. The quantity $N|m|$ is governed by the data size N and the average number of neighbours for each node $|m|$.

The importance of the manipulation above resides in the fact that it allows us to transform the original problem into a linear system of equations. Due to the positive-semidefiniteness of the Hessian H, we can make use of the sparse Cholesky transform proposed in [6] to solve this linear system, whose complexity grows linearly with respect to the number of non-zero entries in the matrix H. The idea is to first factorise H into the product of an upper-triangular matrix Q_h and its transpose Q_h^T. The linear system of equations can then be solved making use of an algebraic manipulation of the upper and lower-triangular matrices. This treatment yields a computationally efficient means for the optimisation of the cost function which compares favourably to traditional iterative equation solvers such as Gauss-Newton methods and the preconditioned conjugate gradient method. This is particularly important for large linear systems such as those arising from image labeling tasks.

4.3 Labeling with Side Information

At this point, its worth noting that, for many real-world applications we encounter ground truth data whose labels are known beforehand. One such example is interactive image segmentation, where region labels are provided in the form of user scribbles are given [1]. Furthermore, in many scopes of application, a portion of the image labels can still be determined using simple heuristics based on partial optimality. Thus, while doing inference on the label fields, it is desired to keep these predefined labels unchanged. The graph regularisation framework we have discussed so far does not explicitly enforce this constraint. To this end, in this section we develop an extension of the method presented earlier incorporating side information for label inference.

Let Γ denote the set of labeled graph nodes with fixed label vectors $\{y_1, \ldots, y_m\}$. The set of

unlabelled nodes is then given by $\mathcal{U}$. In this manner, we can write the cost function

$$f(X) = \sum_{i \in \mathcal{U}} \sum_{a} C_i (x_{i,a} - \hat{x}_{i,a})^2 + \sum_{\substack{i \sim j \\ i,j \in \mathcal{U}}} w_{i,j}(a,b)(x_{i,a} - x_{j,b})^2 + 2g(X) \tag{14}$$

$$g(X) = \sum_{\substack{i \sim j \\ i \in \mathcal{U}, j \in \Gamma}} w_{i,j}(a,b)(x_{i,a} - y_{j,b})^2$$

In the functional above, the first two terms on the right-hand-side are the same ones as in Equation 4. The only difference with respect to Equation 4 is in the parabolic term $g(X)$. It is obvious that $g(X)$ is convex and, hence, the cost function above is still convex. This is since the sum of convex functions is convex. As a result, we can minimise the new cost function in a similar fashion to that employed previously.

We commence by recovering the partial derivative of $g(X)$ with respect to $x_{i,a}$, which is given by

$$\frac{\delta g(X)}{\delta x_{i,a}} = 2 \sum_{i \sim j} w_{i,j}(a,b)(x_{i,a} - y_{j,b}) \tag{15}$$

In the sake of simplicity, we omit hereafter the conditions $i \in \mathcal{U}$ and $j \in \Gamma$ in the equation above. To take our analysis further, we define the N-dimensional vector $V^{(a)}$ whose ith element is given by $\sum_{i \sim j} \sum_b w_{i,j}(a,b)$. We also employ the shorthand $Z^{(a)}$ whose ith element is given by $\sum_{i \sim j} \sum_b w_{i,j}(a,b) y_{j,b}$. Let $\mathrm{vec}(V)$ and $\mathrm{vec}(Z)$ be two large vectors formed by the concatenation of $V^{(a)}$ and $Z^{(a)}$, respectively. We can then represent g as a function of $\mathrm{vec}(X)$ and, as a consequence, the gradient of $g(\mathrm{vec}(X))$ can be written in compact form as follows

$$\nabla g(\mathrm{vec}(X)) = 2 diag(\mathrm{vec}(V))(\mathrm{vec}(X) - \mathrm{vec}(Z)) \tag{16}$$

By combining Equation 16 with Equation 13, we obtain the linear system of equations for the cost function defined in Equation 14. This is

$$H \mathrm{vec}(\overline{X}) = d \tag{17}$$

$$H = F^T (L + \tilde{C}) F + 2 diag(\mathrm{vec}(V))$$

$$d = u^T \tilde{C} F + u^T L F - \mathrm{vec}(\hat{X})^T \tilde{C} F + 2 diag(\mathrm{vec}(V)) \mathrm{vec}(Z)$$

Again, this is a sparse linear system and can be solved efficiently via Cholesky factorisation.

5 Discrete Label Recovery

In this section we discuss the problem of recovering discrete label assignments from continuous label values obtained from the solution of the optimisation problem in 4 and provide a step sequence for our algorithm. A straightforward way of recovering discrete label assignments is majority voting. This approach often yields satisfactory results. Unfortunately, the main drawback of majority voting resides in the fact that many pixels may have no clear "tendency" with respect to their continuous label vectors. This can be interpreted as the case in which there is no element in the continuous label vector with a large margin over the hypotheses for the preferred label as yielded by the continuous optimisation results. In this case, the choice of labeling tends to a random process. On the other hand, since the energy function is smooth, we can give a high degree of confidence to those labels with large margins over the alternatives. Hence we can fix the labels for those pixels as side information and recover the labels of the pixels that remain making use of a local inference process.

By considering the fact that we are working on lattice-like graphs defined on the images under study with pairwise neighbourhood relations between pixels, we can make use of the following partial optimality criterion for any unlabeled pixel i in the graph

$$x_{i,a} = \begin{cases} 1 & a = \arg\min_a c_i(a) + \sum_{j \sim i} \sum_b v_{i,j}(a,b) x_{j,b} \\ 0 & otherwise \end{cases} \tag{18}$$

We can apply the above partial optimality rule to each unlabeled pixel in any permutation order until all pixels are labeled. Note that different permutations of unlabeled pixels lead to different local minima obtained. Thus, we start from those unlabeled pixels adjacent to the labeled pixels and iterate outwards until all labels are resolved. An example of the results yielded by this process is shown in Figure 1. The initial label map obtained by assigning labels to clear "winners", i.e. those pixels whose continuous label-vector element has a clear margin over the others. The intermediate and final results for the discrete labels recovered by employing partial optimality are displayed in the middle and right-hand panels, respectively. In the figure, black and white regions depict background and foreground labellings, whereas the shade of gray depict unlabeled regions.

We now turn our attention to the step sequence for our algorithm. With the theoretical ingredients in previous sections, we summarise our algorithm in Figure 2.

Notice that the initial estimate for the pixel label values $C_i(a)$, which are abstracted as node

attributes, and the edge weights $W_{i,j}(a,b)$ is application dependent. For denoising, $C_i(a)$ is simply the observed discrete pixel value. For segmentation, $C_i(a)$ is the posterior probability of pixel i being assigned to class a, which can be computed, assuming equal priors, via the following Bayes rule

$$C_{i,a} = \frac{p(i|C_a)p(C_a)}{\sum_a p(i|C_a)P(C_a)} = \frac{p(i|C_a)}{\sum_a p(i|C_a)} \tag{19}$$

where $p(i|C_a)$ is the conditional probability of pixel i in class a. This conditional probability can be estimated from features (for example, histograms) of pixels selected as class a exemplars. Here, for purposes of segmentation, unless otherwise stated, we adopt the Potts MRF model. The weights are determined by

$$W(i,j) = \begin{cases} \lambda \exp\big(-\frac{d(i,j)^2}{\sigma_n^2} - \frac{s(i,j)^2}{\sigma_s^2}\big) & \text{If } i \sim j \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

where the parameter λ controls the relative importance of the data term and the smoothness term and $d(i,j)$ and $s(i,j)$ denote the feature and spatial distances between those pixels indexed i and j, respectively. In the equation above, σ_n and σ_s are two bandwidth parameters, where σ_s is taken to be the average contrast between neighbouring pixels in the image. Also, note that the weight scheme, as defined above, takes into account both the spatial and feature consistencies of the label field and is known to preserve contrast information in segmentation applications [1].

6 Discussion

In this section, we provide further interpretation of our labelling approach. We do this since our method is closely related to graph regularisation and diffusion processes. As a result, the discussion presented here provides a link between MRFs, diffusion processes and their associated graphical models and regularisation schemes.

6.1 Graph Regularisation and Diffusion Processes

Recall that the standard interpretation of a discrete MRF is from a probabilistic point of view. The aim is to recover the configuration that maximises the posterior probability of the labeling, where the unitary term $\psi(X_i)$ in Equation 1 is related to the probability $P(X_i|Y_i)$ of label assignment conditional on the data, whereas, the pairwise term $\psi(X_i, X_j)$ is related to the prior probability

$P(X_i|X_j)$. A graphical illustration of the probabilistic view is shown in the left column of Figure 3.

On the other hand, the MAP problem is converted to the energy minimisation one by applying a negative logarithm and relaxation the label hard limit. Let us have a closer look at the energy function of our formulation in Equation 4. For the sake of simplicity, we restrict ourselves to the binary labeling case. It is worth noting in passing that this is done without any loss of generality since the conclusions drawn here also apply to multiclass settings. For the binary case, we consider a scalar label field $\{X_i : i = 1, \ldots, N\}$ where each $X_i \in \{0, 1\}$ is a state variable. Let $v^+_{i,j}$ be the penalty for assigning different labels to a pixel pair i and j and $v^-_{i,j}$ be the penalty for assigning the pair an equivalent label, i.e. both pixels correspond to the same class. With this notation, we can further simplify the energy function as follows

$$\min \sum_{i \sim j} v^+_{i,j}(x_i - x_j)^2 + \sum_{i \sim j} v^-_{i,j}(x_i + x_j - 1)^2 + \sum_i C_i(x_i - \hat{x}_i)^2 \tag{21}$$

where $C_i = c_i(0) + c_i(1)$ and $\hat{x}_i = \dfrac{c_i(0)}{C_i}$.

The gradient of the cost function in Equation 21 with respect to x_i is given by

$$\nabla_{x_i} f(X) = \sum_j 2v^+_{i,j}(x_i - x_j) + \sum_j 2v^-_{i,j}(x_i + x_j - 1) + 2C_i(x_i - \hat{x}_i) \tag{22}$$

where the summation is taken over the labels of pixel-neighbours over the index i. By setting the gradient to 0, and after some simple algebra, we obtain x_i

$$6.1 x_i = \frac{C_i\hat{x}_i + \sum_j v^+_{i,j}x_j + \sum_j v^-_{i,j}(1 - x_j)}{C_i + \sum_j v^+_{i,j} + \sum_j v^-_{i,j}} \tag{23}$$

We can think of Equation as a process, where the label of each pixel is determined by diffusion in two directions. An illustration of the diffusion process is shown in the right-hand panel of Figure 3, where the circular nodes represent the label variables X_i, and the square nodes correspond to the observations at different pixel locations. Note that diffusion takes place externally and independently at each pixel location, as indicated by the vertical links from the square nodes to the associated circular nodes in the figure. This corresponds to the $\hat{x}_i$ term in Equation 6.1. More importantly, diffusion also happens internally between each pair of neighbouring pixels, as indicated by links between circular nodes in the figure. Each circular node propagates its label value to the neighbouring nodes. There are, however, two different types of propagation schemes applied simultaneously. These correspond to the terms with $v^+_{i,j}$ and $v^+_{i,j}$ as given in Equation 6.1. The first

of these terms propagates positive information to the neighbouring pixels, enforcing smoothness in the labeling process. The second term propagates negative information in the local neighbourhood, encouraging adjacent pixels to have dissimilar label values as compared to their own. The final labeling result is achieved when the equilibrium of the diffusion process is reached as a result of the compromise between the terms in the cost function.

6.2 Graph Regularisation and the Random Walker Algorithm

At this point, we also note that our formulation is reminiscent of the random walker segmentation algorithm proposed by Grady [9]. In fact, the random walker algorithm is a special case of our approach. Note that there are two major differences between our approach and the random walker algorithm. Firstly, the random walker algorithm does not make use of the data term and relies solely on the diffusion over labels between neighbouring pixels. We can view a random walker as a graphical structure without vertical links in the right-hand column of Figure 3. Label information must be available for some pixels to guide the label propagation process. This is since, otherwise, equilibrium would be equivalent to a uniform distribution over the pixel-label values, i.e. all the pixel label-values become the same. Hence, the random walker can only be applied to the supervised learning scenario. Our approach, on the other hand, incorporates the data term in the cost function and can be applied to unsupervised cases, as long as the data and binary potentials can be effectively computed. Secondly, the random walker algorithm only propagates positive information at each pixel location and, hence, limits itself to enforcing smoothness between neighbouring pixels. Although this is adequate for many applications such as image denoising and segmentation, it is still preferable to have a more general framework which enables us to accommodate other types of pairwise constraints. Further, strictly speaking, the random walker is not a relaxation over the discrete MRF in general, but rather a relaxation of the Potts MRF model, which is a special type of MRF subject to smoothness constraints.

In passing, we also note that, in machine learning research, similar techniques based on graphs have been proposed for solving semi-supervised learning problems [23, 22]. Indeed, the labeling problem in vision and the semi-supervised learning problem in machine learning can both be thought as the generic problem of inference over structured data linked through the general framework of MRF. Like the random walker algorithm, the work in [23, 22] also considers label smoothness and can be viewed as variants of Potts MRF model relations.

6.3 L_2 vs. L_1 Optimisation

Along the discussion above, it is important to note that the choice of norm in Equation 4 is not unique. By utilising the L-1 norm instead of the L-2 Euclidean norm employed before, we can obtain a different cost function given by

$$\begin{aligned}\min f(X) = &\sum_{i=1}^{N}\sum_{a=1}^{K}|c_i(a) - x_{i,a}| \\ &+ \sum_{i\sim j}\sum_{a=1}^{K}\sum_{b=1}^{K} w_{i,j}(a,b)|x_{i,a} - x_{j,b}|\end{aligned} \tag{24}$$

By introducing auxiliary slack variables $t_{i,a}$ for each node indexed i with label a and binary relations $v_{i,a,j,b}$ for the node pair i, j and with labels a and b, we can reformulate Equation 24 in terms of the following linear program

$$\min\ f(X,T,V) = \sum_{i,a} t_{i,a} + \sum_{i,a,j,b} v_{i,a,j,b} \tag{25}$$

$$s.t.\ \begin{cases} |x_{i,a} - c_i(a)| \leq t_{i,a} \forall i, a \\ |x_{i,a} - x_{j,b}| \leq u_{i,a,j,b} \forall i, a, j, b \\ \sum_a x_{i,a} = 1\ \forall i \text{ and } x_{i,a} \geq 0\ \forall i, a \\ t_{i,a} \geq 0\ \forall i, a \text{ and } v_{i,a,j,b} \geq 0\ \forall i, a, j, b \end{cases}$$

Compared to the quadratic programming (QP) formulation presented in previous sections, the above LP relaxation has a higher computational complexity. This is particularly evident for graphs with large number of nodes. Also note that, in the case of the L_1 norm, the constraints must be explicitly taken into account. Moreover, recent work in continuous relaxation-based MRF inference [14, 17] suggests second order relaxation methods provide a margin of improvement in performance over linear programming relaxation. Consequently in the sake of efficiency and accuracy, in our experiments we focus in the use of our convex quadratic formulation.

7 Experimental Results

In this section, we demonstrate the utility of our method for purposes of image labeling. To this end, we commence by showing segmentation results of the proposed method on synthetic images with various levels of Gaussian noises. We then turn our attention to real-world colour imagery.

For the real-world image data, we provide results for both, the binary, i.e. foreground-background segmentation, and the multiclass settings. Finally, we show results of our algorithm on multispectral image labeling as applied to remote sensing material classification.

In our experiments, for segmentation of real images, we adopted the optimal parameter setting for graph cuts based segmentation as reported in [20]. For segmentation of the noisy synthetic data, we only make use of the spatial term in Equation 20 with $\sigma_s = 2$. This implies that, in our experiments on noisy synthetic data, the results are devoid of the parameters λ_2 and σ_s. As mentioned earlier, the parameter λ controls the relative importance of the data term and the smoothness term, which we have set to 10 empirically for experiments on synthetic data and 50 for experiments on real data. For the parameter λ_2, we have used a value of 10 for real data segmentation so as to remove small and isolated areas with high contrast.

7.1 Synthetic Images

We performed two experiments on synthetic imagery. For the first experiment, we generated two images, the first one is comprised by pixels arranged columnwise whose intensities are 0.2, 0.4 and 0.6. The second image depicts a 2×2 checkerboard with intensities ranging from 0.1 to 0.7 in 0.2 intervals, starting from the top-left square in clockwise order. Random gaussian noise with standard deviation of 0.2, equal to the difference of intensity values for neighbouring patches, has been applied to both images, yielding the noise corrupted imagery shown in the leftmost columns of Figure 4. The purpose here is to recover the image labels given the noise statistics, where each label corresponds to an intensity level.

Results recovered by the proposed labeling approach are shown in the right-hand column of Figure 4. In the middle column, we show the results yielded by Maximum Likelihood Estimation (MLE). For the alternative, we have used a Gaussian prior corresponding to the added noise. We have also used the intensities of the patches in clean images to estimate the conditional likelihood. As can be seen from Figure 4, our regularisation approach clearly outperforms MLE, even in cases when the prior and the conditional likelihood are accurate as respect to the noise corruption and image labels.

Next, we continue with our analysis making use of an image with more sophisticated patterns. This image depicts a circle whose normalised intensity is of 0.2 in the middle of three larger, darker squares whose intensities vary from 0.4 to 0.8 in increments of 0.2. As before, we have applied

increasing degrees of Gaussian noise of zero-mean and standard deviation $0.1, 0.2, 0.3, 0.4$ to our test image. For this set of experiments, we have labeled a few pixels from each class and used them as side information to govern the segmentation process. The noisy images, with their respective "brush" labelings, are shown in the top row of Figure 5. For the brush labelings, we have used a mask for the sake of consistency and, in the panels, each colour corresponds to one of the four different classes in the image. The results yielded by our method are shown in the bottom row of Figure 5. The middle row shows the results delivered by MLE. As shown in the figure, our method is quite robust to noise corruption. Even with a standard deviation of 0.4, which is twice as much as the shade of gray difference between regions in clean images.

Moreover, despite large amounts of noise corruption, our method can still yield plausible segmentation results. This is confirmed by the quantitative analysis of the results obtained by both, MLE and our approach. To this end, we generate 10 different noisy images by adding random noise with zero-mean and increasing values of variance. With the images at hand, we apply MLE and our graph regularisation approach to the noisy imagery. The resulting label maps are then compared with the ground truth. The error plots corresponding to the percentage of wrongly labelled pixels as a function of noise variance are shown in the in Figure 6. From the figure, we can conclude that our method performs significantly better than MLE, for which the performance downgrades asymptotically to random guess with increasing levels of noise variance.

7.2 Interactive Segmentation of Real-world Images

We now turn our attention to the interactive segmentation of real-world colour images. To do this, we have employed selected images from the Berkeley Image Database [16]. For each image, a few pixels were selected by the user so as to define the foreground and background side-information regions. The purpose is to assign the unlabelled pixels into predefined regions determined by the labeled pixels. Notice that the user interface of interactive image segmentation can be provided by other means rather than user specified scribble brushes. The GrabCut system [18] provides a different user interface by making use of rectangular shaped bounding boxes to circumscribe the foreground object of interest. This is particularly suited for the case when the foreground occupies a small bounded region in the image. Despite the difference in the user interface, the GrabCut system still used graph cuts to solve the labeling problem that arises from the underlying MRF model as in [1].

For binary foreground-background segmentation, we compared our results with the graph-cut algorithm in [1] as well as the random walker algorithm in [9]. Again, for the sake of consistency, we use the same labeling mask and bandwidth parameters for both, our method and the alternatives. In our experiments, we have used the maximum flow code in [2] for solving the graph cuts and the sparse Cholesky factorisation method in [6] for solving linear equations involved in our QP optimisation.

Sample results on background-foreground, i.e. binary, segmentation are shown in Figure 7. In the sake of clarity of presentation, the original images in the figure have been superimposed with user-provided labels. The results of our algorithm, and those yielded by graph cuts and the random walker are displayed in the left, middle and right-hand columns, respectively. From the segmentation results, we notice that, in a binary labelling setting, our algorithm achieves much better performance than the random walker and is comparable with graph cuts. In our experiments, it is somewhat surprising that the random walker results tend to overflow if the brush labeled pixels are very close to the boundary. Nonetheless, this can be explained by the fact that the random walker algorithm is, as mentioned earlier, a relation on a Potts MRF model with smoothness constraints.

Note that neither our algorithm nor the graph cuts presents such overflow problems. This is due to the control exerted by the color model in the data terms of the cost functions employed by both methods. Our algorithm preserves the structure of the object better than the graph cuts, specially for elongated objects. This is evident in the case of the scissors and the sheep legs on the images in the second and third rows in Figure 7. For these images, graph cuts produce "shortcut" effects in segmentation even with refined labeling as indicated by white dots in the images. Our method, on the other hand, does not have such problems. For the other images we have tested, our method produces quite consistent results as compared to the results yielded by graph cuts. We did not find a single instance for which graph cuts can achieve significantly better segmentation quality than ours. In regards to efficiency, our method has a low computational cost. For an image with spatial resolution $640x480$, our method takes 3 seconds on average for labeling.

Also, note that our method can accommodate arbitrary pairwise terms and is not restricted to submodular potential functions [13]. This is since it operates on the label-set like other continuous relaxation methods. Thus, in our method, submodularity is not a condition for optimality. Our method can also handle multi-class segmentation in a straightforward manner. To illustrate this, we also show results on multi-class image segmentation in Figure 8. From the figures, we can conclude that the method is capable of recovering segmentation results that capture the structure of

the regions labeled using the brush. As in Figure 7, our method enforces within-region smoothness while preserving the the fine detail of the segmented objects.

7.3 Multispectral Image Labeling

Our labeling approach can also be applied to a broader area of applications on different feature sets. To illustrate this, we show an example of our method applied to multispectral imagery. More specifically, we aim at performing material classification via labeling pixels on a hyperspectral image captured through remote sensing. In a multispectral image, each pixel is associated with a spectral signal, which facilitates the discrimination of different materials based on their distinct spectral features. As a result, we can pose the material identification problem as a classification one with predefined number of classes. For each class, a few pixels have been randomly selected as initial seeds. As an alternative to our method, we have applied Linear Discriminant Analysis (LDA) as a preprocessing step to reduce the dimensionality of raw spectra and build an Gaussian classifier upon the extracted LDA features. For the sake of consistency, we have used the seeds employed by our algorithm as the labeled training samples for the LDA.

For our method, the initial estimate of label values can be computed via Equation 19, Note that, for the alternative, we can view the posterior $p(i|C_a)$ in Equation 19 as the output of individual Gaussian classifiers. Thus, whereas the alternative can be viewed as a MAP estimation process, whereas our approach is a regularisation one over the initial posterior values. The labeling results of both, our approach and the alternative, are shown in Figure 9. In the figure, we also show the pseudocolour image, i.e. the image for which the red-green-blue channels are given by three bands in the input image, and the ground truth map. We can see that our regularisation-based labeling approach not only achieves a much higher accuracy rate of 93.23% as compared to the alternative, which yields 79.02% accuracy, but also produces a more visually desirable result as compared to the ground truth.

8 Conclusions

In this paper, we have presented an image labelling approach based upon graph regularisation. The multiclass image labeling method we have presented here is closely related to the Markov Random Field model. We have recast inference of labels into a continuous optimisation setting over the label

fields. The method is based upon the extremisation of a convex cost function which arises from a relaxation process on a MRF. Moreover, the cost function is a quadratic one with sparse Hessian matrix. Hence the minimisation can be efficiently done by solving a sparse linear system. We have also discussed the links of our method to diffusion processes and elaborated further on the choice of minimisation strategy. We have illustrated the utility of our labeling algorithm for purposes of segmentation of synthetic and real-world imagery and material identification on multispectral images.

A Proof of Extrema Equivalence Between the MRF Negative Log-likelihood and our Cost Function

In this Appendix, we provide a proof for the equivalence between the negative log-likelihood function for the MRF model as presented in Equation 3 and our alternative energy functional introduced in Equation 4. More specifically, we aim at showing that

$$\begin{aligned} f(X) &= \sum_{i=1}^{N}\sum_{a=1}^{K}\gamma_a(X_i) + \sum_{i\sim j}\sum_{a=1}^{K}\sum_{b=1}^{K} w_{i,j}(a,b)(x_{i,a}-x_{j,b})^2 \\ &= -\log P(X) = \sum_{i=1}^{N}\sum_{a=1}^{K} c_i(a)x_{i,a} + \sum_{i\sim j}\sum_{a=1}^{K}\sum_{b=1}^{K} v_{i,j}(a,b)x_{i,a}x_{j,b} + C \\ &= q(X) + C \end{aligned} \tag{26}$$

where C is a constant independent of the label values and the variables $\gamma_a(X_i)$, $w_{i,j}(a,b)$, $c_i(a)$ and $v_{i,j}(a,b)$ are as given in Sections 3 and 4. In other words, we aim at showing that both functionals above are equivalent up to an additive constant, which is independent of the optimisation variables.

To this end, we proceed in a two-step fashion. Firstly, we consider the unitary terms in both equations. Consider an arbitrary node indexed i with label $X_i = a$, where $x_{i,k} = 1$ for $k = a$ and $x_{i,k} = 0$ otherwise. Hence, we have

$$\begin{aligned} f(X_i) &= \sum_{k}\gamma_k(X_i) \\ &= (k-1)\sum_{k=1}^{K} c_i(k) + c_i(a) = q(X_i) + \sum_{k=1}^{K} c_i(k) \end{aligned} \tag{27}$$

where $\sum_k c_i(k)$ is a constant independent of the variable $x_{i,k}$.

Secondly, we take a similar approach regarding the binary terms in Equations 3 and 4. Consider two neighbouring nodes indexed i and j whose labels are given by $X_i = a$ and $X_j = b$, respectively. Thus, by setting $\gamma_k(X_i) = 0$ and $c_i(k) = 0$ for all i and k, we have

$$\begin{aligned} f(X_i, X_j) &= \sum_{k=1}^{K}\sum_{k'=1}^{K} w_{i,j}(k,k')(x_{i,k} - xj, k')^2 = \sum_{k\neq b} w_{I,j}(a,k) + \sum_{k\neq a} w_{i,j}(k,b) \\ &= \sum_{k\neq b}\sum_{a'\neq a}\sum_{b'\neq k} v_{i,j}(a',b') + \sum_{k\neq a}\sum_{a'\neq k}\sum_{b'\neq b} v_{i,j}(a',b') \\ &= \sum_{a'=1}^{K}\sum_{b'=1}^{K} v_{i,j}(a',b') + v_{i,j}(a,b) = q(X_i, X_j) + \sum_{a'=1}^{K}\sum_{b'=1}^{K} v_{i,j}(a',b') \end{aligned} \tag{28}$$

were, again, the term $\sum_{a'}\sum_{b'} v_{i,j}(a',b')$ is a constant independent of $x_{i,a'}$.

As a result the extrema of both, $f(X)$ and $q(X)$ are equivalent. This is due to the fact that, $f(X) = q(X) + C$, where the constant additive factor C is given by

$$C = \sum_{i=1}^{N}\sum_{k=1}^{K} c_i(k) + \sum_{i\sim j}\sum_{a'=1}^{K}\sum_{b'=1}^{K} v_{i,j}(a',b') \tag{29}$$

References

[1] Y. Boykov and M-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Intl. Conf. on Computer Vision*, pages 105–112, 2001.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans.on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[4] Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[5] T. Cour and J. Shi. Solving markov random fields with spectral relaxation. In *Intl. Conf. on Artificial Intelligence and Statistics*, 2007.

[6] T. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, 2006.

[7] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000.

[8] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[9] L. Grady. Random walks for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2006.

[10] E. R. Hancock and J. V. Kittler. Discrete relaxation. *Pattern Recognition*, 23:711–733, 1990.

[11] J. Keuchel. Multiclass image labeling with semidefinite programming. In *European Conference on Computer Vision*, pages 454–467, 2006.

[12] J. Keuchel, C. Schnorr, C. Schellewald, and D. Cremers. Binary partitioning, perceptual grouping, and restoration with semidefinite programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(11):1364–1379, 2003.

[13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via grph cuts? *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[14] M.P. Kumar, P.H.S. Torr, and A. Zisserman. Solving markov random fields using second order cone programming relaxations. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1045–1052, 2006.

[15] S.Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 2001.

[16] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Int. Conf. on Computer Vision*, volume 2, pages 416–423, July 2001.

[17] P. Ravikumar and J. Lafferty. Quadratic programming relaxations for metric labeling and markov random field map estimation. In *Intl. Conf. on Machine Learning*, pages 737–744, 2006.

[18] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics*, 23(3):309–314, 2004.

[19] C. Rother, V. Kolmogorov, V. S. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

Figure 1: Illustration of Discrete Label Recovery. Left Column: initial label map yielded by the hard limit of the continuous output; Middle Column: intermediate result; Right Column: final result yielded by using partial optimality.

[20] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *European Conf. on Computer Vision*, pages 16–29, 2006.

[21] P.H.S. Torr. Solving markov random fields using semi definite programming. In *Intl Workshop on Artificial Intelligence and Statistics*, 2003.

[22] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Neural Information Processing Systems*, 2003.

[23] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *20th Intl. Conf. on Machine Learning*, 2003.

1 Compute the likelihood values $C_i(a)$ for individual nodes and determine initial weights between neighbouring nodes and labels $W_{i,j}(a, b)$.

2 Construct H and d via Equation 13. Note that, for labeling with hard constraints, Equation 17 should be used.

3 Solve the linear system $H\mathrm{vec}(X) = d$ via sparse Cholesky factorisation.

4 Recover the continuous label estimates by reshaping $\mathrm{vec}(\overline{X})$ and abutting the last column following the sum-to-one constraint.

5 Recover the discrete labels from the continuous estimates using the criteria provided in this section.

Figure 2: Graph Regularisation based Labeling Algorithm

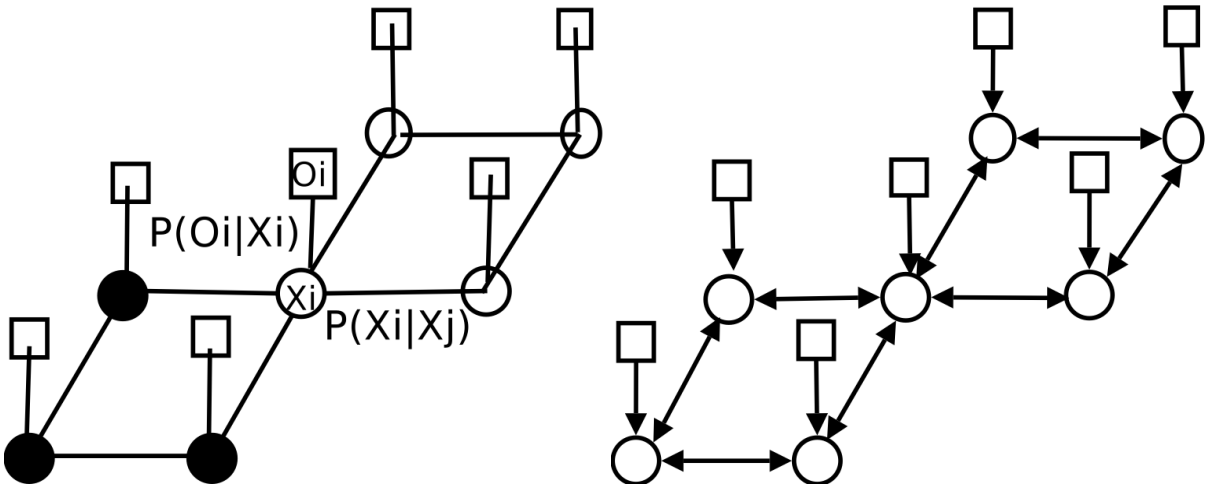

Figure 3: Two graphical interpretations of Pairwise Discrete Markov Random Fields. Left-hand panel: the probabilistic modeling view; Right-hand panel: The model corresponding to the diffusion viewpoint.

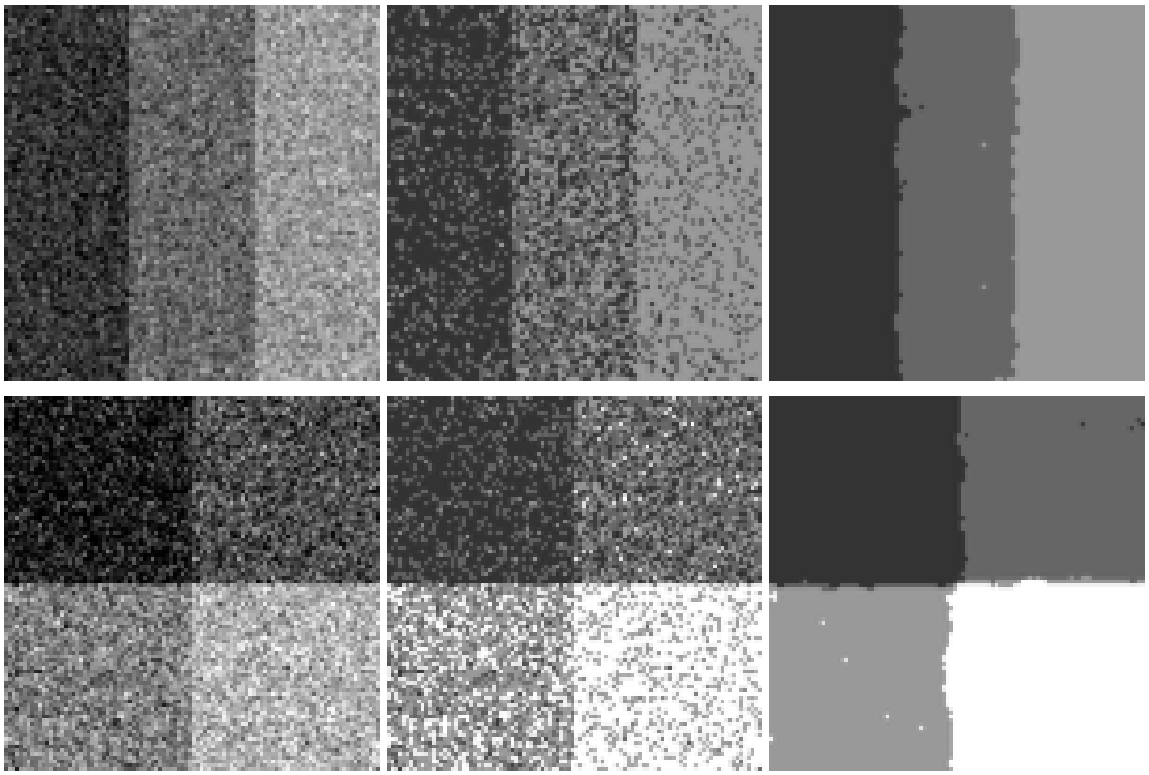

Figure 4: Synthetic image labeling results. The columns show, from left-to-right, the input noisy images, results obtained making use of MLE and those yielded by our approach.

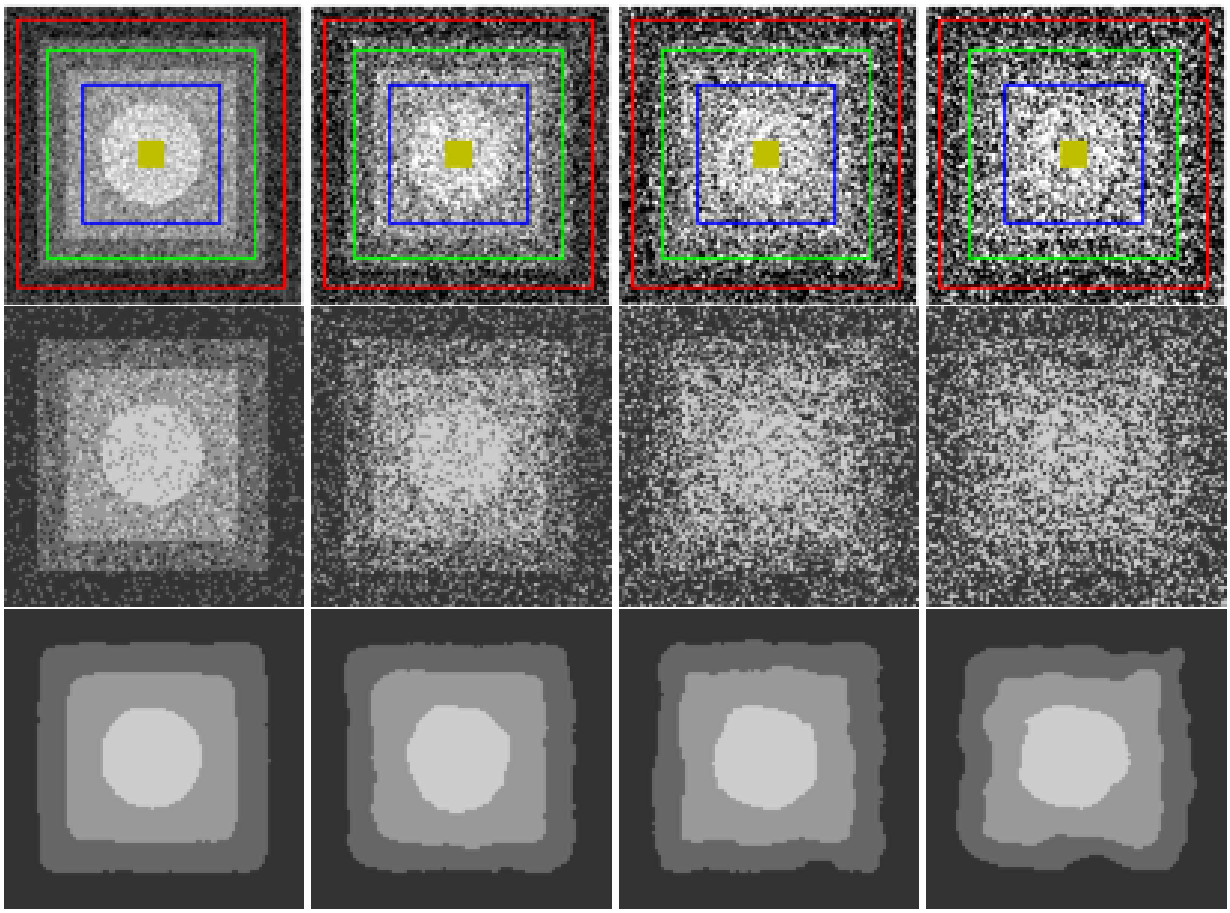

Figure 5: Top row: input images with increasing levels of Gaussian noise. Middle row: Maximum Likelihood Estimation (MLE) Results. Bottom row: Results of our regularisation approach.

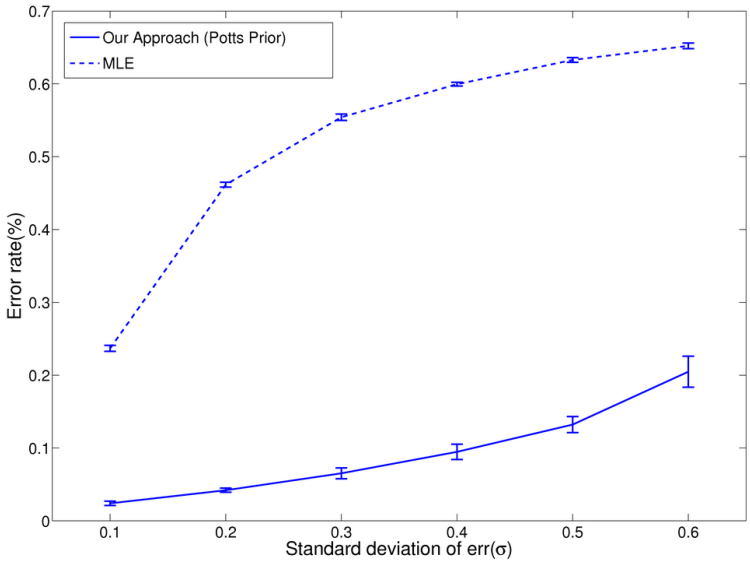

Figure 6: Performance of our method and MLE as a function of noise standard deviation for the noisy synthetic imagery in Figure 5.

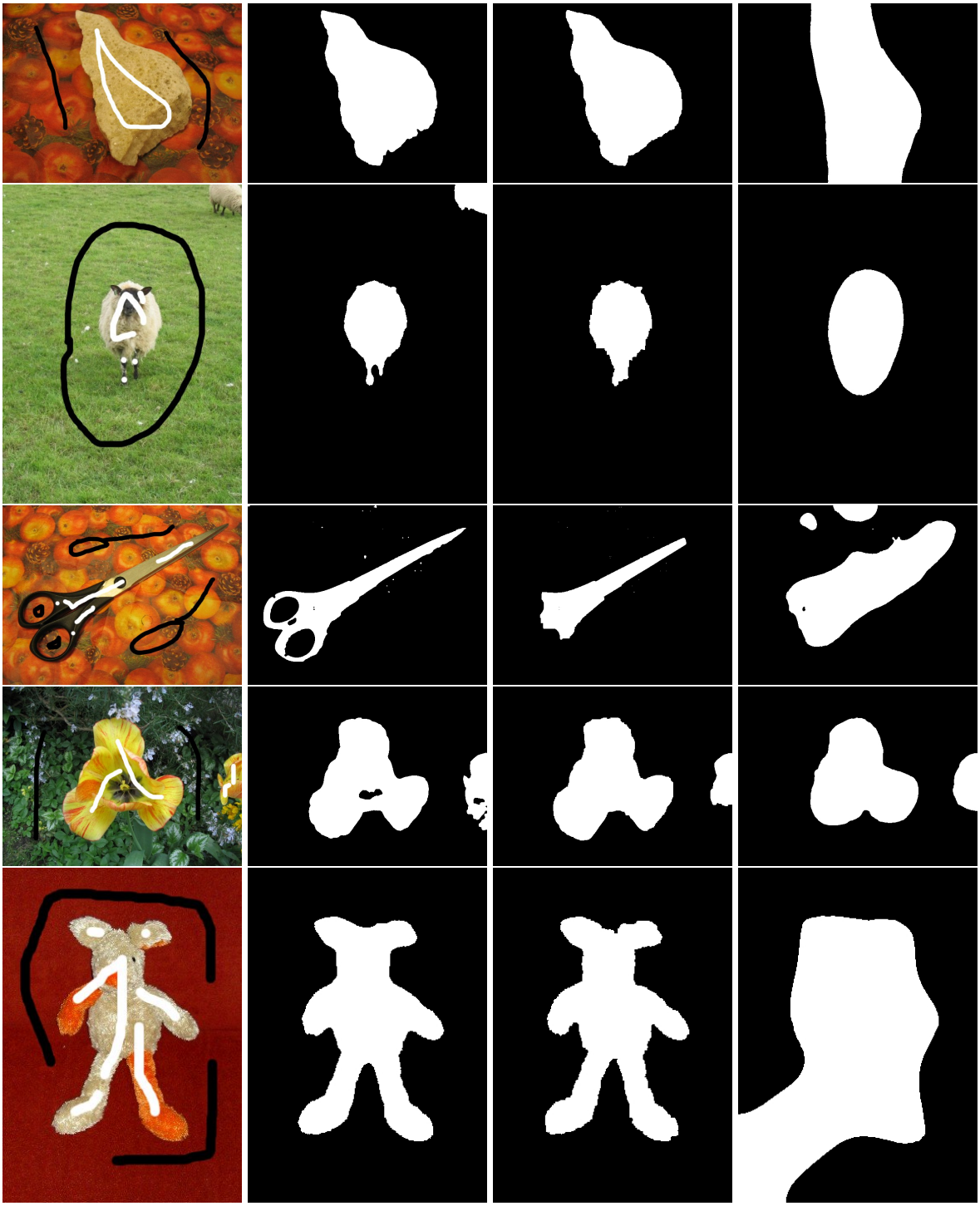

Figure 7: Examples of binary foreground/background segmentation. From left-to-right: Input images with brush labelings, results yielded by our algorithm, result yielded by graph cuts [1], labelling delivered by the random walker algorithm[9].

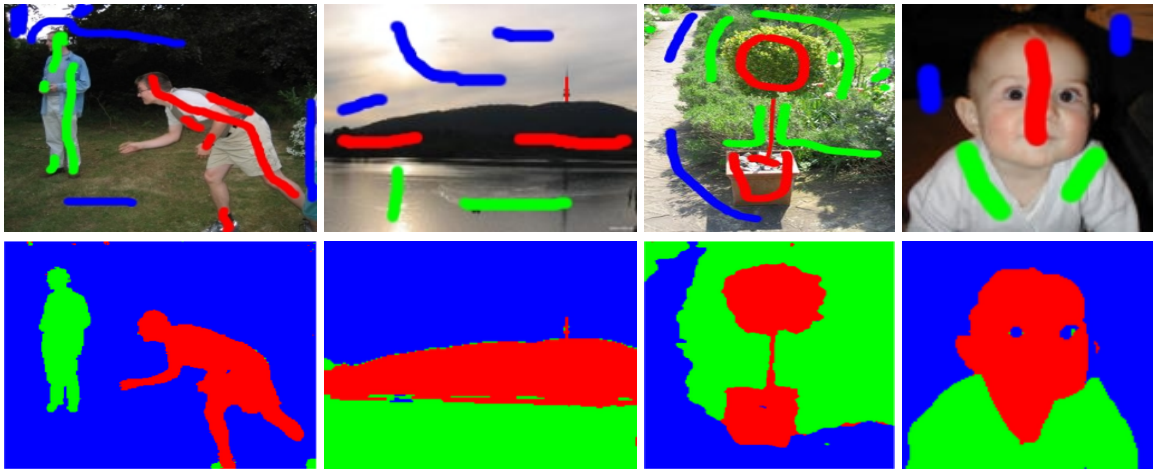

Figure 8: Examples of multi-class image segmentation. Top row: Input images with brush labelings; Bottom row: Results recovered by our approach

Figure 9: Results of multispectral image labeling. The images, from left-to-right, correspond to the pseudo color image, growth truth label map, LDA classifier results (with an accuracy rate of 79.02%) and our result (with an accuracy rate of 93.23%).