Solving Markov Random Fields using Second Order Cone Programming Relaxations

M.Pawan Kumar P.H.S. Torr
Dept. of Computing
Oxford Brookes University
{pkmudigonda,philiptorr}@brookes.ac.uk
http://cms.brookes.ac.uk/computervision

A. Zisserman
Dept. of Engineering Science
University of Oxford
az@robots.ox.ac.uk
http://www.robots.ox.ac.uk/~vgg

Abstract

*This paper presents a generic method for solving Markov random fields (*MRF*) by formulating the problem of* MAP *estimation as 0-1 quadratic programming (*QP*). Though in general solving* MRF*s is* NP*-hard, we propose a second order cone programming relaxation scheme which solves a closely related (convex) approximation. In terms of computational efficiency, our method significantly outperforms the semidefinite relaxations previously used whilst providing equally (or even more) accurate results. Unlike popular inference schemes such as Belief Propagation and Graph Cuts, convergence is guaranteed within a small number of iterations. Furthermore, we also present a method for greatly reducing the runtime and increasing the accuracy of our approach for a large and useful class of* MRF*s. We compare our approach with the state-of-the-art methods for subgraph matching and object recognition and demonstrate significant improvements.*

1 Introduction

Many Computer Vision problems, from stereo reconstruction and segmentation to object category recognition, can be given a probabilistic formulation using Markov random fields (MRF). An MRF is described by a set of sites along with a neighbourhood relationship on them. Each site i has a label x_i chosen from a discrete or continuous set of labels, $\mathcal{X}_i$. Associated with each possible configuration $\mathbf{x}$ of the MRF is its posterior probability $\Pr(\mathbf{x})$.

The problem of *solving an* MRF or *performing inference* is that of finding the maximum a-posteriori (MAP) configuration. Special forms of MRFs can be efficiently solved in polynomial time. For example, tree-structured MRFs and *k-fans* [4] can be solved using Belief Propagation [13] and certain binary MRFs can be solved using Graph Cuts [6]. However, in general, this problem is NP-hard.

Several approximate inference algorithms have been proposed in the literature. Bayesian methods such as Loopy Belief Propagation and Generalized Belief Propagation [17] build upon message passing algorithms proposed for tree-structured MRFs and attempt to minimize Bethe and Kikuchi approximations of the free energy respectively. Although some success has been reported using these methods for certain problems [12], they are not guaranteed to converge. Moreover, the computational complexity typically increases exponentially with the number of sites per *region* of the free energy approximation [17].

Energy minimization techniques such as $\alpha\beta$-swap and α-expansion [2], popular for low-level vision problems, solve MRFs as a series of binary labelling problems using Graph Cuts. However, they are not guaranteed to converge in a finite number of iterations. They also require the set of labels $\mathcal{X}_i$ to be the same for all sites i, which might require introducing additional labels per site thereby making the algorithm computationally expensive.

Recently, the inference problem for MRFs has been shown to be equivalent to an NP-hard 0-1 quadratic programming (QP) problem whose convex approximation can be solved using semidefinite relaxations [15]. Although this method has the desirable property of convergence, it is practically infeasible for two reasons: (i) methods for solving semidefinite programming (SDP) are in general inefficient; (ii) it squares the number of variables making inference even more expensive.

We propose a method for efficiently solving the QP problem using relaxations based on second-order cone programming (SOCP). Our approach takes inspiration from the relaxations proposed for a similar NP-hard problem called MAX-CUT [11]. Advantages of our method are five-fold: (i) the algorithm does not place any restrictions on the structure of the MRF, the pairwise potentials or the labels $\mathcal{X}_i$; (ii) it is guaranteed to converge within a small number of iterations; (iii) it uses efficient interior-point algorithms for solving the resulting SOCP [9]; (iv) it uses fewer variables than the SDP approach [15]; (v) for a large and very useful class of MRFs,

it further allows us to greatly reduce the number of variables by exploiting the structure of the problem.

In section 2, we briefly outline the 0-1 quadratic programming formulation for the problem of solving MRFs. Section 3 describes SOCP relaxations for general quadratic problems. In section 4 we provide details of a new SOCP relaxation for solving MRFs. In Section 5 we propose a speed-up of our approach for a large class of MRFs. Results of our method and a comparison with the state-of-the-art methods for two challenging problems, subgraph matching and object recognition, are presented in section 6.

2 Quadratic Programming Formulation for Solving Markov Random Fields

In this section, we briefly review Markov random fields and provide a 0-1 quadratic programming formulation for solving them.

2.1 Markov Random Fields

A Markov random field (MRF) is described by a set of n_S sites $\{1, 2, ..., n_S\}$ along with a symmetric neighbourhood relationship $N(.)$ on them, i.e. $i \in N(j)$ if and only if $j \in N(i)$. Each site i has a label $x_i \in \mathcal{X}_i$. We assume that the sets $\mathcal{X}_i$ are finite and discrete ($|\mathcal{X}_i| = n_L < \infty$). Without loss of generality, we assign $\mathcal{X}_i = \{1, 2, \cdots, n_L\}$. Associated with each configuration $\mathbf{x}$ of the MRF is its posterior probability $\Pr(\mathbf{x})$. According to the Hammersley-Clifford theorem,

$$\Pr(\mathbf{x}) = \frac{1}{Z} \prod_i \exp(-\phi_i(x_i)) \prod_{ij} \exp(-\phi_{ij}(x_i, x_j)). \quad (1)$$

Here Z is a normalization constant, $\exp(-\phi_i(x_i))$ is the likelihood of a site i having a label x_i and $\exp(-\phi_{ij}(x_i, x_j))$ is the prior for two neighbouring sites i and j having labels x_i and x_j. The terms $\phi_i(x_i)$ and $\phi_{ij}(x_i, x_j)$ are called the unary and pairwise potentials of the MRF respectively. The problem of solving an MRF is to find a configuration $\mathbf{x}^*$ such that

$$\begin{aligned} \mathbf{x}^* &= \arg\max_{\mathbf{x}} \Pr(\mathbf{x}), \\ &= \arg\min_{\mathbf{x}} E(\mathbf{x}), \end{aligned} \quad (2)$$

where $E(\mathbf{x}) = \sum_i \phi_i(x_i) + \sum_{ij} \phi_{ij}(x_i, x_j)$ is the energy of the MRF. Note that Z does not appear in the energy as it is a constant for all possible configurations of the MRF. Next, we describe the 0-1 quadratic programming formulation of the above problem.

2.2 Quadratic Programming Formulation

In order to formulate the problem of solving an MRF as 0-1 quadratic programming we introduce three new set of variables.

Binary vector m: We define a binary vector $\mathbf{m}$ of size $n_S n_L$ such that $\mathbf{m}_{x_i + n_L*(i-1)} = 1$, if site i has the label x_i and $\mathbf{m}_{x_i + n_L*(i-1)} = 0$, otherwise. Since each site i has a unique label, we have the constraint $\sum_{x_i} \mathbf{m}_{x_i + n_L*(i-1)} = 1$ (i.e. sum over all possible values of label x_i), $\forall i$.

Likelihood vector l: The likelihood vector $\mathbf{l}$ is also a vector of size $n_S n_L$ and is defined as $\mathbf{l}_{x_i + n_L*(i-1)} = \phi_i(x_i)$. For any configuration $\mathbf{x}$, $\sum_i \phi_i(x_i) = \mathbf{m}^T \mathbf{l}$.

Prior matrix P: The $n_S n_L \times n_S n_L$ prior matrix $\mathbf{P}$ is defined as $\mathbf{P}_{x_i + n_L*(i-1), x_j + n_L*(j-1)} = \phi_{ij}(x_i, x_j)$ if i and j are neighbours, and 0 otherwise. For any configuration $\mathbf{x}$, $\sum_{ij} \phi_{ij}(x_i, x_j) = \frac{1}{2} \mathbf{m}^T \mathbf{P} \mathbf{m}$.

Quadratic Programming. Clearly, the problem of solving an MRF, described in equation (2), is equivalent to the following 0-1 quadratic programming (QP):

$$\begin{aligned} \mathbf{m}^* &= \arg\min_{\mathbf{m}} \mathbf{m}^T \mathbf{l} + \frac{1}{2} \mathbf{m}^T \mathbf{P} \mathbf{m}, \\ \text{subject to } & \sum_{x_i=1}^{n_L} \mathbf{m}_{x_i + n_L*(i-1)} = 1, \forall i. \end{aligned} \quad (3)$$

For a general non-binary MRF whose neighbourhood relationship $N(.)$ does not define a tree structure, the above QP is NP-hard. However, we propose a method which solves a closely related convex approximation based on second-order cone programming (SOCP) relaxations. We begin by describing SOCP in the next section.

3 Second-order Cone Programming

In this section, we describe second-order cone programming (SOCP) and SOCP relaxations for general quadratic problems. A second-order cone (also called quadratic, Lorentz or ice-cream cone) in d dimensions is defined as $\|\mathbf{u}\| \leq t$ where $\mathbf{u} \in \Re^{d-1}$ and $t \in \Re$. Note that a second-order cone defines a convex set. An SOCP is a problem with a linear objective function whose feasible set (i.e. the set of variables satisfying all the constraints) is given by the intersection of a finite number of (affinely deformed) second-order cones. Since the intersection operation preserves convexity, SOCP falls under the category of convex optimization. Formally, an SOCP is a problem of the following form:

$$\begin{aligned} \min \quad & \mathbf{f}^T \mathbf{y} \\ \text{subject to} \quad & \|\mathbf{u}_i\| \leq t_i, i = 1, \cdots, L, \\ & \mathbf{u}_i = \mathbf{A}_i \mathbf{y} + \mathbf{b}_i, i = 1, \cdots, L, \\ & t_i = \mathbf{c}_i^T \mathbf{y} + d_i, i = 1, \cdots, L. \end{aligned} \quad (4)$$

Here, $\mathbf{y} \in \Re^n$ is the variable to be optimized, $\mathbf{f} \in \Re^n$ is the optimization function, and the second-order cone constraints are specified by $\mathbf{A}_i \in \Re^{n_i - 1 \times n}$, $\mathbf{b}_i \in \Re^{n_i - 1}$, $\mathbf{c}_i \in \Re^n$, and $\mathbf{d}_i \in \Re$. Note that we introduce additional variables $\mathbf{u}_i \in \Re^{n_i - 1}$ and $t_i \in \Re$ to convert the constraints $\|\mathbf{A}_i \mathbf{y} + \mathbf{b}_i\| \leq \mathbf{c}_i^T \mathbf{y} + d_i$ into the standard form

of a second-order cone. SOCP can be regarded as a special case of semidefinite programming (SDP) since $\|\mathbf{u}\| \leq t$ is equivalent to the constraint $\begin{pmatrix} tI & \mathbf{u} \\ \mathbf{u}^T & t \end{pmatrix} \succeq 0$, where I is an identity matrix of appropriate dimensions and $\mathbf{X} \succeq \mathbf{Y}$ means that $\mathbf{X} - \mathbf{Y}$ is positive semidefinite.

An SOCP of the above form can be solved efficiently using interior-point algorithms in $O(\sqrt{L})$ iterations, each of complexity $O(n^2 \sum_i n_i)$. In comparison, SDP takes $O(\sqrt{\sum_i n_i})$ iterations of complexity $O(n^2 \sum_i n_i^2)$. In practice, SOCP has been found to greatly outperform similar SDP problems in terms of computational efficiency [11].

Several important convex optimization problems, such as convex quadratically constrained quadratic programming, sum of norms and its special case of L_1 minimization, maximum of norms and its special case of L_∞ minimization [5] as well as problems with hyperbolic constraints can be formulated as SOCPs. See [9] for further details.

Many important problems in Computer Vision, however, are non-convex, i.e. the feasible set of these problems is not convex. Thus, they cannot be solved directly using SOCP. One could attempt solving them using general non-linear programming techniques such as Levenberg-Marquardt. However, these have the undesirable property of requiring an initial point which greatly influences the accuracy of the result. Thus, often these methods get trapped in a local minimum.

A more popular approach is to *relax* the original non-convex problem by loosening some constraints such that the feasible set becomes convex. Although this is an approximation, an appropriate relaxation results in an easy-to-solve convex optimization problem whose solution is close to the optimal solution of the original (non-convex) problem, e.g see [11]. In this paper, we are concerned with relaxations which can be solved using SOCP.

SOCP Relaxations for Quadratic Problems. Consider the following QP problem:

$$\begin{aligned} \min \quad & \mathbf{y}^T \mathbf{Q}_0 \mathbf{y} + \mathbf{q}_0^T \mathbf{y} \\ \text{subject to} \quad & \mathbf{y}^T \mathbf{Q}_p \mathbf{y} + \mathbf{q}_p^T \mathbf{y} + \gamma_p \leq 0, p = 1, \cdots, m, \end{aligned} \quad (5)$$

where $Q_i \in S(n)$ are symmetric $n \times n$ matrices (not necessarily positive semidefinite, thus making the problem non-convex). The QP in equation (3) is one instance of such a problem. Let $\mathbf{Y} = \mathbf{y}\mathbf{y}^T$, then the above problem can be reformulated as

$$\begin{aligned} \min \quad & -\gamma_0 \\ \text{subject to} \quad & \mathbf{Q}_p \bullet \mathbf{Y} + \mathbf{q}_p^T \mathbf{y} + \gamma_p \leq 0, p = 0, \cdots, m, \\ & \mathbf{Y} = \mathbf{y}\mathbf{y}^T, \end{aligned} \quad (6)$$

where the operator ($\bullet$) defines an element-wise multiplication, i.e. $\mathbf{X} \bullet \mathbf{Y} = \sum_{ij} x_{ij} y_{ij}$. Note that in the above problem, we introduce two additional variables $\mathbf{Y}$ and γ_0 which need to be optimized together with the original variable $\mathbf{y}$. This problem is NP-hard because the constraint $\mathbf{Y} = \mathbf{y}\mathbf{y}^T$ describes a non-convex set. This is illustrated in Fig. 1 for the case where $\mathbf{y}$ is a 1-dimension vector (i.e. a scalar). This constraint can be relaxed (loosened) in several ways such that the feasible set becomes convex.

LP Relaxation. If the constraint is simply ignored, the problem is reduced to linear programming (which is convex). This is known as *lift-and-project* (LP) or *reformulation-linearization* technique. Although such a relaxation results in excellent speed-ups, the quality of the solution obtained is often unsatisfactory.

SDP Relaxation. Another approach is to use homogeneous coordinates for $\mathbf{y}$ thereby obtaining the *semidefinite (SDP) relaxation* $\begin{pmatrix} 1 & \mathbf{y}^T \\ \mathbf{y} & \mathbf{Y} \end{pmatrix} \succeq 0$, instead of $\mathbf{Y} = \mathbf{y}\mathbf{y}^T$. The above relaxed constraint is equivalent to $\mathbf{Y} \succeq \mathbf{y}\mathbf{y}^T$ and describes a convex set, i.e. the cone of semidefinite matrices and thus the problem becomes convex. The resulting semidefinite program (SDP) is computationally inefficient but provides more accurate results than LP relaxation. A similar approach was used to solve MRFs in [15].

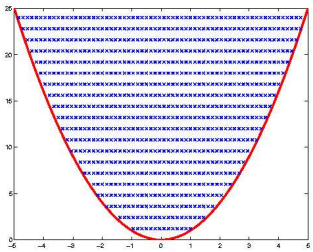


Figure 1. *The red curve shows the non-convex set specified by the constraint $\mathbf{Y} = \mathbf{y}\mathbf{y}^T$ when $\mathbf{y}$ is scalar, i.e. $\mathbf{Y} = \mathbf{y}^2$. The shaded region lying above the curve shows the convex set defined by the SDP relaxation $\mathbf{Y} \succeq \mathbf{y}\mathbf{y}^T$ of the constraint. Note that in this case the SOCP relaxation $\mathbf{y}\mathbf{y}^T \bullet \mathbf{S} \leq \mathbf{Y} \bullet \mathbf{S}$ is equivalent to the SDP relaxation since all 1×1 positive semidefinite matrices are multiples of each other.*

SOCP Relaxation. These relaxations lie somewhere between LP and SDP relaxations. Though they do not ignore the constraint $\mathbf{Y} = \mathbf{y}\mathbf{y}^T$ completely, SOCP relaxations specify only a subset of constraints obtained using the SDP relaxation $\mathbf{Y} \succeq \mathbf{y}\mathbf{y}^T$. Thus, SOCP relaxations are more efficient than SDP relaxations and more accurate than LP relaxations. For sparse matrices $\mathbf{Q}_i$, they often outperform SDP relaxation by allowing more constraints to be specified [11].

We now describe a particular type of SOCP relaxation. Suppose we are given a set of positive semidefinite matrices $\mathcal{S}$. The choice of the set $\mathcal{S}$ differs from one problem to the other. We will specify the set $\mathcal{S}$ which we use for solving MRFs in the next section. Since any $\mathbf{S} \in \mathcal{S}$ is positive semidefinite, it can be decomposed as $\mathbf{S} = \mathbf{U}\mathbf{U}^T$. If

$\mathbf{Y} \succeq \mathbf{y}\mathbf{y}^T$, then

$$\begin{aligned}(\mathbf{Y} - \mathbf{y}\mathbf{y}^T) \bullet \mathbf{S} &= (\mathbf{Y} - \mathbf{y}\mathbf{y}^T) \bullet \mathbf{U}\mathbf{U}^T \\ &= \sum_i \mathbf{u}_i^T (\mathbf{Y} - \mathbf{y}\mathbf{y}^T)\mathbf{u}_i \geq 0, \quad (7)\end{aligned}$$

where $\mathbf{u}_i$ are the rows of the matrix $\mathbf{U}$. As we will see, the above constraint defines a convex set and in fact, can be written as an SOCP constraint as follows:

$$\begin{aligned}&\mathbf{y}\mathbf{y}^T \bullet \mathbf{S} \leq \mathbf{Y} \bullet \mathbf{S} \\ \Rightarrow\ &\mathbf{y}^T \mathbf{U}\mathbf{U}^T \mathbf{y} \leq \mathbf{Y} \bullet \mathbf{S} \\ \Rightarrow\ &\|\mathbf{U}^T \mathbf{y}\|^2 \leq \mathbf{Y} \bullet \mathbf{S} \quad (8)\end{aligned}$$

Equation (8) is equivalent to the following SOCP constraint:

$$\left\| \begin{pmatrix} 1 - \mathbf{Y} \bullet \mathbf{S} \\ 2\mathbf{U}^T \mathbf{y} \end{pmatrix} \right\| \leq 1 + \mathbf{Y} \bullet \mathbf{S}. \quad (9)$$

This can be derived by squaring both sides of equation (9) and cancelling out the common terms (i.e. $1 + (\mathbf{Y} \bullet \mathbf{S})^2$). *Thus, we relax the non-convex constraint $\mathbf{Y} = \mathbf{y}\mathbf{y}^T$ to the set of convex constraints $\mathbf{y}\mathbf{y}^T \bullet \mathbf{S} \leq \mathbf{Y} \bullet \mathbf{S}$, for all $\mathbf{S} \in \mathcal{S}$.* For the case where $\mathbf{y}$ is a scalar, the constraint defined in equation (9) is the same for all 1×1 positive semidefinite matrices $\mathbf{S}$. The convex set defined by such a constraint is shown in Fig. 1.

If $\mathcal{S}$ is the set of all positive semidefinite matrices, then the SOCP relaxation is equivalent to the SDP relaxation. However, we choose $\mathcal{S}$ as a subset of positive semidefinite matrices appropriate for our problem as described in the next section. Note that the variable to be optimized in the original problem in equation (5) was of dimension $n_S n_L$. By introducing $\mathbf{Y}$ in equation (6), we have squared the number of variables, i.e. the dimensionality of the problem is increased to $O(n_S^2 n_L^2)$. However, using the structure of the problem, this is greatly reduced as described in § 4.1.

4 Solving MRFs using SOCP

We now describe how to choose the appropriate set of semidefinite matrices $\mathcal{S}$ for the problem. Let $\mathbf{y} = 2\mathbf{m} - \mathbf{1}$ where $\mathbf{1}$ is a one vector of dimension $n_S n_L$, and $\mathbf{Y} = \mathbf{y}\mathbf{y}^T$. Since $\mathbf{m}$ is a binary vector, $\mathbf{y}$ is a vector whose elements are either -1 or 1. In § 4.1 we describe how the constraint $Y_{ii} = 1$ helps in reducing the number of variables. Our problem in equation (3) can now be reformulated as

$$\begin{aligned}\mathbf{y}^* &= \arg\min_{\mathbf{y}} \mathbf{y}^T (2\mathbf{l} + \mathbf{P}\mathbf{1}) + \frac{1}{2}\mathbf{P} \bullet \mathbf{Y}, \\ \text{subject to } &\sum_{x_i} \mathbf{y}_{x_i + n_L * (i-1)} = 2 - n_L, \forall i. \quad (10)\end{aligned}$$

When choosing the set $\mathcal{S}$, we want to include a large set of positive semidefinite matrices thereby ensuring that the relaxation is good. Recall that if all positive semidefinite matrices were included in $\mathcal{S}$ then we would have the SDP relaxation. As an example, consider the case where $n_S n_L = 2$ and the set $\mathcal{S}$ which consists of the following semidefinite matrices:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}. \quad (11)$$

Implicitly, the convex constraint $\mathbf{y}^T \mathbf{S} \mathbf{y} \leq \mathbf{Y} \bullet \mathbf{S}$ must be satisfied for any positively weighted linear combination of the above matrices, i.e. any $\mathbf{S}$ of the form $\begin{pmatrix} a+c+d & c-d \\ c-d & b+c+d \end{pmatrix}$, where $a, b, c, d \geq 0$. This includes a large subset of 2×2 positive semidefinite matrices and hence the resulting relaxation is good. However, it does not include all the 2×2 semidefinite matrices (i.e. where a diagonal element is less than the off-diagonal elements). Thus, the SOCP relaxation specifies only a subset of the constraints obtained using SDP relaxation. Note that empirically we found this choice of set $\mathcal{S}$ to work well. However, other choices may also result in a good relaxation. We now generalize the set $\mathcal{S}$ for any n_S and n_L. We will then describe how this choice of $\mathcal{S}$ enables us to reduce the number of variables.

Let $\mathbf{e}_j \in \Re^{n_S n_L}$ be a zero vector except for the j-th component, which is 1. We define vectors $\mathbf{u}_{ij} = \mathbf{e}_i + \mathbf{e}_j$ and $\mathbf{v}_{ij} = \mathbf{e}_i - \mathbf{e}_j$. We also define the set $\mathcal{E}$ as

$$\begin{aligned}&\mathcal{E} = \{(i', j')\}, \\ &\text{such that } \exists i, j \in \{1, 2, .., n_S\}, i \in N(j), \\ &\exists x_i \in \mathcal{X}_i, \exists x_j \in \mathcal{X}_j, \\ &i' = x_i + n_L * (i - 1), \\ &j' = x_j + n_L * (j - 1). \quad (12)\end{aligned}$$

In other words, the set $\mathcal{E}$ consists of all pairs of binary variables which belong to neighbouring sites of the MRF. Following the relaxation in [11] for MAX-CUT, we choose $\mathcal{S}$ as:

$$\begin{aligned}&\mathbf{e}_i \mathbf{e}_i^T, i = 1, \cdots, n_S n_L, \\ &\mathbf{u}_{i'j'} \mathbf{u}_{i'j'}^T, (i', j') \in \mathcal{E}, \\ &\mathbf{v}_{i'j'} \mathbf{v}_{i'j'}^T, (i', j') \in \mathcal{E}. \quad (13)\end{aligned}$$

For the above $\mathcal{S}$, the corresponding convex constraints are:

$$\mathbf{y}^T \mathbf{e}_i \mathbf{e}_i^T \mathbf{y} \leq \mathbf{Y} \bullet \mathbf{e}_i \mathbf{e}_i^T, \quad (14)$$

$$\mathbf{y}^T \mathbf{u}_{i'j'} \mathbf{u}_{i'j'}^T \mathbf{y} \leq \mathbf{Y} \bullet \mathbf{u}_{i'j'} \mathbf{u}_{i'j'}^T, \quad (15)$$

$$\mathbf{y}^T \mathbf{v}_{i'j'} \mathbf{v}_{i'j'}^T \mathbf{y} \leq \mathbf{Y} \bullet \mathbf{v}_{i'j'} \mathbf{v}_{i'j'}^T. \quad (16)$$

It is desirable to eliminate $\mathbf{Y}$, which squares the number of variables, from the above inequalities and substitute it with a smaller number of variables. We now show that our choice of the set $\mathcal{S}$ makes this possible in our method.

4.1 Reducing the number of variables

We define the following variables:

$$s_{i'j'} \equiv \mathbf{Y} \bullet \mathbf{u}_{i'j'}\mathbf{u}_{i'j'}^T,$$
$$z_{i'j'} \equiv \mathbf{Y} \bullet \mathbf{v}_{i'j'}\mathbf{v}_{i'j'}^T. \quad (17)$$

The variables $s_{i'j'}$ are indicators with value 0 and 4 which are turned on when $y_{i'}$ and $y_{j'}$ have the same sign. Similarly the indicator variables $z_{i'j'}$ are turned on when $y_{i'}$ and $y_{j'}$ have different signs. Together they help us to pick out the appropriate pairwise potential (i.e. $\phi_{i'j'}(x_{i'}, x_{j'})$) from the matrix $\mathbf{P}$. Similar variables have been used in TRW [10, 16] (an LP relaxation of equation (3)) and SDP relaxations [15]. Note that the number of such variables is $O(|\mathcal{E}|)$ which is much less than $O(n_S^2 n_L^2)$ variables introduced by $\mathbf{Y}$. Using $s_{i'j'}$ and $z_{i'j'}$, it is possible to eliminate $\mathbf{Y}$ as follows.

Proposition 1. The constraints in equations (14)-(16) can be reformulated as

$$y_i^2 \leq 1,$$
$$(\mathbf{y}_{i'} + \mathbf{y}_{j'})^2 \leq s_{i'j'},$$
$$(\mathbf{y}_{i'} - \mathbf{y}_{j'})^2 \leq z_{i'j'},$$
$$s_{i'j'} + z_{i'j'} = 4. \quad (18)$$

Proof in appendix. Observe that the constraint $s_{i'j'} + z_{i'j'} = 4$ offers a strict lower bound on $s_{i'j'}$ and $z_{i'j'}$ thereby avoiding trivial solutions where these variables can assume arbitrarily large values. In addition, we must also eliminate $\mathbf{Y}$ from the objective function of equation (10). For this purpose, consider the following proposition.

Proposition 2. For any prior matrix $\mathbf{P}$, $\mathbf{P} \bullet \mathbf{Y} = -\sum_{i'j'} \mathbf{P}_{i'j'} z_{i'j'}$. Proof in appendix.

Therefore, the SOCP relaxation of the problem in equation (10) is given by:

$$\begin{aligned} \mathbf{y}^* = \quad & \arg\min_{\mathbf{y}} \mathbf{y}^T(2\mathbf{l} + \mathbf{P}\mathbf{1}) - \tfrac{1}{2}\textstyle\sum_{i'j'} \mathbf{P}_{i'j'} z_{i'j'}, \\ \text{subject to} \quad & \mathbf{y}_i^2 \leq 1, i = 1, \cdots, n_S n_L \\ & (\mathbf{y}_{i'} + \mathbf{y}_{j'})^2 \leq s_{i'j'}, (i', j') \in \mathcal{E}, \\ & (\mathbf{y}_{i'} - \mathbf{y}_{j'})^2 \leq z_{i'j'}, (i', j') \in \mathcal{E}, \\ & s_{i'j'} + z_{i'j'} = 4, (i', j') \in \mathcal{E}, \\ & \textstyle\sum_{x_i} \mathbf{y}_{x_i + n_L * (i-1)} = 2 - n_L, i = 1, \cdots, n_S. \end{aligned} \quad (19)$$

There are two types of relaxations involved in the above problem: (i) the constraint $\mathbf{y}_i \in \{-1, 1\}$ is relaxed to $\mathbf{y}_i \in [-1, 1]$; and (ii) the constraint $\mathbf{Y} = \mathbf{y}\mathbf{y}^T$ is relaxed to the set of constraints $\mathbf{y}^T\mathbf{S}\mathbf{y} \leq \mathbf{S} \bullet \mathbf{Y}$, for all $\mathbf{S} \in \mathcal{S}$. The above problem is convex and can be formulated as an SOCP as described in equation (9) and solved efficiently using interior-point algorithms [9].

Note that the number of variables have been reduced from $O(n_S^2 n_L^2)$ to $O(|\mathcal{E}|)$ due to the substitution of $\mathbf{Y}$ by variables $s_{i'j'}$ and $z_{i'j'}$. For small or sparse MRFs, the above problem becomes computationally feasible. In the case of grid graphs, e.g. an image, $|\mathcal{E}| \approx 2n_s n_L^2$. This leads to a reduction of $O(n_S)$ where n_S are the number of pixels in the image. For example, for a standard 640×480 images, $n_S = 307200$. In the next section, we will describe a method for further reducing the number of variables for a large class of MRFs.

4.2 Triangular Inequalities

Other than the constraints specified in equation (19), we also use the *triangular inequalities* described below. As in the case of MAX-CUT [11], we found that they greatly improve the accuracy of the results.

For any three elements $\mathbf{y}_i, \mathbf{y}_j$ and $\mathbf{y}_k$ of the vector $\mathbf{y}$, at least two of them will have the same sign. Therefore, the following inequalities hold true:

$$\begin{aligned} \mathbf{Y}_{ij} + \mathbf{Y}_{jk} + \mathbf{Y}_{ik} &\geq -1, \\ \mathbf{Y}_{ij} - \mathbf{Y}_{jk} - \mathbf{Y}_{ik} &\geq -1, \\ -\mathbf{Y}_{ij} - \mathbf{Y}_{jk} + \mathbf{Y}_{ik} &\geq -1, \\ -\mathbf{Y}_{ij} + \mathbf{Y}_{jk} - \mathbf{Y}_{ik} &\geq -1. \end{aligned} \quad (20)$$

The variable $\mathbf{Y}$ can be eliminated from the above inequalities by reformulating it as follows.

Proposition 3. If $(i, j), (j, k), (k, l) \in \mathcal{E}$, then the above constraints can be restated as

$$\begin{aligned} z_{ij} + z_{jk} + z_{ik} &\leq 8, \\ z_{ij} + s_{jk} + s_{ik} &\leq 8, \\ s_{ij} + s_{jk} + z_{ik} &\leq 8, \\ s_{ij} + z_{jk} + s_{ik} &\leq 8. \end{aligned} \quad (21)$$

Proof in appendix. Note that the triangular inequalities can also be specified in SDP relaxations. However, this makes the problem computationally infeasible due to the complexity of the algorithms for solving SDP. In the next section, we describe an approach to further reduce the number of variables for a large and useful class of MRFs thereby making our approach tractable for even non-sparse MRFs.

5 Robust Truncated MRF

This section describes a class of MRFs which we call the *robust truncated* MRF (due to the special form of their pairwise potentials $\phi_{ij}(x_i, x_j)$). We also describe an efficient SOCP relaxation for solving them. The pairwise potentials of such an MRF form a robust truncated model as follows:

$$\begin{aligned} \phi_{ij}(x_i, x_j) &= f_{ij}(x_i, x_j), \text{ if } x_i \in \mathcal{C}_i(x_j), \\ &= \tau_{ij}, \text{ otherwise}, \end{aligned} \quad (22)$$

where $\mathcal{C}_i(x_j)$ are the subset of labels of i which are *compatible* with x_j. In other words, the pairwise potential for all incompatible pair of labels is truncated to τ_{ij} and

thus, this MRF is robust to the presence of outliers. Included in this class are the commonly used Potts model i.e. $f_{ij}(x_i, x_j) = d_{ij}, \forall x_i, x_j$, the truncated linear model i.e. $f_{ij}(x_i, x_j) = |x_i - x_j|$ and the truncated quadratic model i.e. $f_{ij}(x_i, x_j) = (x_i - x_j)^2$. In most problems, $f_{ij}(x_i, x_j) < \tau_{ij}$ and the number of labels n_C in $\mathcal{C}_i(x_j)$ are much smaller than n_L, i.e. $n_C \ll n_L$. Such MRFs have been used extensively in various vision applications, from stereo reconstruction [6] to object category segmentation [8].

We now present an SOCP relaxation for solving robust truncated MRFs which greatly reduces the number of variables. The prior matrix $\mathbf{Q}$ (different from the previous prior matrix $\mathbf{P}$) for such an MRF is defined as

$$\begin{aligned}\mathbf{Q}_{x_i+n_L*(i-1),x_j+n_L*(j-1)} &= \phi_{ij}(x_i, x_j) - \tau_{ij}, \text{if } i \in N(j),\\ &= 0, \text{otherwise.}\end{aligned}$$

Since $\sum_{x_i} \mathbf{y}_{x_i+n_L*(i-1)} = 2 - n_L, \forall i$, we get $\mathbf{y}^T\mathbf{Q}\mathbf{y} = \mathbf{y}^T\mathbf{P}\mathbf{y} + \text{constant}$. Note that the matrix $\mathbf{Q}$ is very sparse since $n_C \ll n_L$. Thus, the number of variables are greatly reduced from $O(|\mathcal{E}|)$ to $O(n_C n_S^2)$. The SOCP relaxation for equation (10) is as follows:

$$\begin{aligned}\mathbf{y}^* = \quad & \arg\min_{\mathbf{y}} \mathbf{y}^T(2\mathbf{l} + \mathbf{P1}) - \tfrac{1}{2}\textstyle\sum_{i'j'} \mathbf{Q}_{i'j'} z_{i'j'},\\ \text{subject to} \quad & \mathbf{y}_i^2 \le 1, i = 1, \cdots, n_S n_L\\ & (\mathbf{y}_{i'} + \mathbf{y}_{j'})^2 \le s_{i'j'}, Q_{i',j'} \neq 0,\\ & (\mathbf{y}_{i'} - \mathbf{y}_{j'})^2 \le z_{i'j'}, Q_{i',j'} \neq 0,\\ & s_{i'j'} + z_{i'j'} = 4, Q_{i',j'} \neq 0,\\ & \textstyle\sum_{x_i} \mathbf{y}_{x_i+n_L*(i-1)} = 2 - n_L, i = 1, .., n_S \quad (23)\end{aligned}$$

Like equation (19), the above problem can be converted into an SOCP using equation (9) and solved efficiently using interior-point algorithms [9].

In addition to the constraints in equations (21) and (23), one might want to specify *compatibility* constraints, i.e. ensure that the labels of two neighbouring sites i and j are compatible ($\phi_{ij}(x_i, x_j) < \tau_{ij}$). The (approximation of) pairwise potential $\phi_{ij}(x_i, x_j)$ is given by $-\frac{1}{2}\sum_{i'j'} \mathbf{Q}_{i'j'} z_{i'j'}$ where the variables $\mathbf{y}_{i'}$ and $\mathbf{y}_{j'}$ correspond to sites i and j respectively. Thus, the compatibility constraint is given by $\sum_{i'j'} \mathbf{Q}_{i'j'} z_{i'j'} > 0$. which greatly improves the accuracy of the result at the cost of specifying one additional linear constraint. We now present results of our method for subgraph matching and object recognition.

6 Results

We now formulate two important problems using the robust truncated model. This allows us to use our efficient SOCP relaxation described in equation (23) (solved using MOSEK [1]). The first is the problem of subgraph matching using synthetic data, where ground truth is known. The second problem is object recognition using real image data. We compare our approach with the state-of-the-art methods for both problems and demonstrate significant improvements.

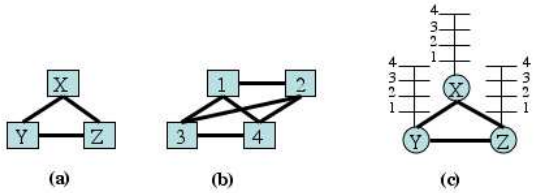


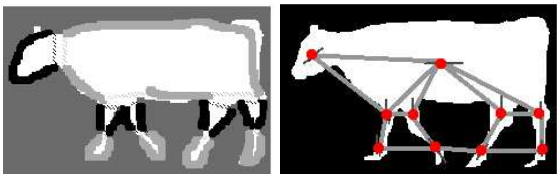
Figure 2. *Subgraph Matching. (a) Graph $\mathcal{G}_1$ with three vertices which is a rigidly transformed subgraph of graph $\mathcal{G}_2$ shown in (b). (c) The corresponding MRF formulation for subgraph matching. The MRF consists of three sites corresponding to the vertices X,Y and Z of $\mathcal{G}_1$. Each site has four possible labels corresponding to vertices 1,2,3 and 4 of $\mathcal{G}_2$. The interactions between the sites is shown using solid lines.*

Synthetic Experiment: Subgraph Matching. We use SOCP relaxation to solve the problem of subgraph matching. Given two graphs $\mathcal{G}_1 = \{\mathcal{V}_1, \mathcal{E}_1\}$ and $\mathcal{G}_2 = \{\mathcal{V}_2, \mathcal{E}_2\}$, subgraph matching involves finding a mapping $f : \mathcal{V}_1 \rightarrow \mathcal{V}_2$ which minimizes the energy function $\sum_{v_i, v_j \in \mathcal{V}_1} \|l_{ij}^1 - l_{f(i)f(j)}^2\|$, where l_{ij}^k is the distance between vertices i and j of the k^{th} graph. Many important Computer Vision problems, such as matching part-based models for object recognition can be thought of as special cases of this problem.

We define an MRF for determining the mapping $f(.)$ such that each site i represents a vertex v_i^1 in $\mathcal{V}_1$. Each label x_i represents a vertex v_i^2 in $\mathcal{V}_2$. For our example, we assume that all points $v_i^1 \in \mathcal{V}_1$ are equally likely to map to a point in $\mathcal{V}_2$, and hence the likelihood terms $\exp(-\phi_i(x_i))$ are set to 0.5 (however this is not generally the case). The sites of the MRF form a complete graph as distances between all pairs of vertices should be preserved by the mapping. We define the pairwise potentials $\phi_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ as d_1 if $|l_{ij}^1 - l_{x_i x_j}^2| \le \epsilon$ and d_2 otherwise, where ϵ is a constant which depends on the (expected) level of noise. In our experiments, we use $d_1 = 0.1$ and $d_2 = 1.0$. This favours the preservation of distance between corresponding pairs of vertices. Figure 2 shows an example of this formulation when $|\mathcal{V}_1| = 3$ and $|\mathcal{V}_2| = 4$.

Our problem reduces to obtaining the MAP estimate given the above MRF. For this purpose, we use three variants of the SOCP relaxation. SOCP-A uses only the relaxation specified in equation (23) without specifying any triangular inequalities, described in equation (21), or compatibility constraints. SOCP-B uses 400 randomly selected triangular inequalities, while SOCP-C also specifies the compatibility constraints. We compare our results with LP and previously used SDP relaxations [14, 15] as well as the state-of-the-art method in subgraph matching. Note that though additional constraints can be specified when using SDP relaxations, they increase the time complexity of the algorithm considerably, and are thus not included.

We generated 1000 pairs of random graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, with $|\mathcal{V}_1| = 0.25|\mathcal{V}_2|$ on an average. The number of vertices $|\mathcal{V}_2|$ were varied between 20 and 30. The vertices $|\mathcal{V}_1|$ were



(a) (b)

Figure 3. *Example layered pictorial structure (LPS) of a cow. (a) Various parts p_i of the cow (head, torso and half-limbs) (b) A PS can be viewed as an MRF. The sites are shown as red circles. The black lines show the location and orientation of the parts and the grey lines show a subset of the connections between parts. Note that the connections actually form a complete graph.*

randomly selected subset of $|\mathcal{V}_2|$ with 5% noise added. On an average, we found $n_C n_S^2 = 0.1|\mathcal{E}|$ which allowed us to greatly reduce the number of variables as described in section 5. The accuracy of matches for the vertices in $\mathcal{V}_1$ found using SOCP-C were 96.18% (6225 out of 6472) compared to 6.64% (430 out of 6472) found using LP and 93.11% (6026 out of 6472) using SDP.

Table 1 summarizes the average time and accuracy of the various methods used. By allowing additional constraints, SOCP-C provides more accurate results than SDP and is significantly more efficient. Note that though the LP relaxation is faster than SOCP, its accuracy is very poor. We also outperformed the state-of-the-art method by Chui and Rangarajan [3] (tested using their publically available code) on this challenging dataset. Note that due to the large amount of noise, the distances from two nearby vertices v_1^i and v_1^j to any other vertex v_1^k can be very similar. In other words, the pairwise potentials provide insufficient information for determining the mapping $f(.)$. Hence, the accuracy of our algorithm, or indeed any algorithm, does not reach 100%.

Method	Time	Accuracy (%)
LP	0.85 sec	6.64
SDP [15]	35.0 sec	93.11
SOCP-A	3.0 sec	92.01
SOCP-B	4.5 sec	94.79
SOCP-C	4.8 sec	96.18
[3]	2.2 sec	20.00

Table 1. *Average time and accuracy for subgraph matching.*

Real-data Experiment: Object Recognition We tested our approach for binary object present/absent recognition test, i.e. to determine whether a given image contains an instance of a particular object category or not. For this purpose, we use our parts-based model called layered pictorial structures (LPS) [7]. Note that this was the underlying model for obtaining object category specific segmentation in [8].

LPS is an MRF whose sites correspond to parts such that the LPS would provide a generative model for the object (see [8]). The parts and the graph structure of the LPS of a cow are shown in Fig. 3. Each site i of the MRF, shown as a red circle in Fig. 3(b), represents a part of the object. Each label x_i of site i represents a putative pose $\mathbf{l}_i = \{px_i, py_i, \theta_i, \rho_i\}$ of the part where $\{px_i, py_i\}$ is the position, θ_i is the rotation and ρ_i is the scale. In order to model the likelihood $\exp(-\phi_i(x_i))$ of each label, we use two types of features for the shape and appearance of the part (see [8] for details).

The connections between the parts of the LPS form a complete graph. A subset of the pairwise connections are shown in Fig. 3(b) in grey lines. The pairwise potentials are chosen to be a robust truncated model and are given by

$$\phi_{ij}(x_i, x_j) = \begin{cases} d_1 & \text{if valid configuration,} \\ d_2 & \text{otherwise.} \end{cases}$$

A configuration is said to be valid if $x_{ij}^{min} \leq \|x_i - x_j\| \leq x_{ij}^{max}$, where the limits x_{ij}^{min} and x_{ij}^{max} are learnt using the training data as described in [8].

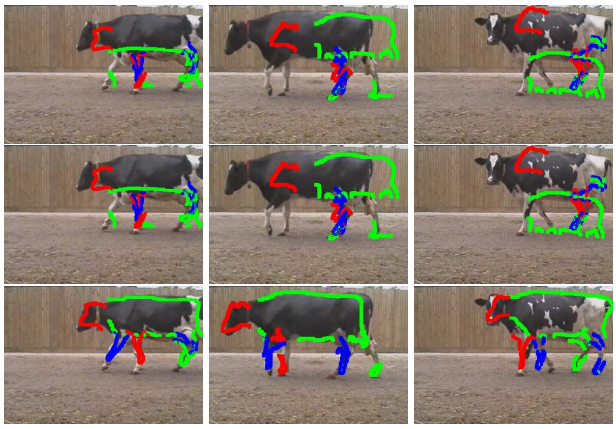


Figure 4. *Results of matching LPS to images for object recognition. The outline of the parts detected are shown overlaid on the images. The first row shows the results obtained using LBP run over 50 iterations. Parts detected using GBP, again using 50 iterations, are shown in the second row. Note that both these methods do not converge for the above images. The results obtained using SOCP relaxations, which are guaranteed to converge, are shown in the third row. Clearly, they outperform the other two methods.*

The problem of object recognition now reduces to that of solving the MRF corresponding to the LPS model. We found on an average $n_C n_S^2 = 0.05|\mathcal{E}|$. This allows us to efficiently solve the MRF using SOCP relaxation as described in section 5. Our algorithm takes approximately 2 minutes on a 2.4 GHz machine to find the MAP estimate for each image. We compared our approach with Loopy Belief Propagation (LBP) and Generalized Belief Propagation (GBP) using a maximum of 50 iterations. For testing, we used 450 positive and 2400 negative examples which were not present in the training set. LBP and GBP do not converge for 16% and 10% of the images respectively. Fig. 4 shows the advantage of our method over LBP and GBP in these cases. We observed that increasing the number of iterations for LBP and GBP did not improve the results. The ROC curves shown in

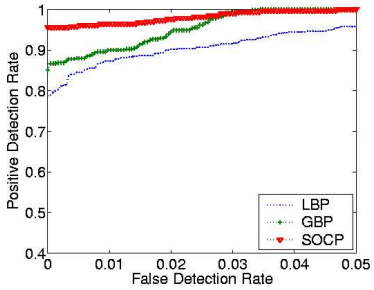


Figure 5. ROC *curves for cow recognition. Results indicate that the* SOCP *relaxation outperforms* LBP *and* GBP.

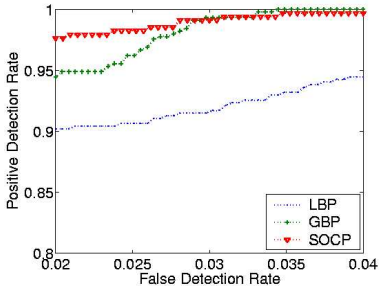


Figure 6. *Zoomed versions of a part of the* ROC *curve.*

Fig. 5 and 6 demonstrate that we significantly outperform LBP and GBP.

Appendix

Proof of Proposition 1. Using the fact that $Y_{ii} = 1$, equation (14) reduces to $\mathbf{y}_i^2 \leq 1$. By definition of $s_{i'j'}$ and $z_{i'j'}$, we get $(\mathbf{y}_{i'} + \mathbf{y}_{j'})^2 \leq s_{i'j'}$ and $(\mathbf{y}_{i'} - \mathbf{y}_{j'})^2 \leq z_{i'j'}$. Finally, we observe that $s_{i'j'} + z_{i'j'} = \mathbf{Y} \bullet (\mathbf{u}_{i'j'}\mathbf{u}_{i'j'}^T + \mathbf{v}_{i'j'}\mathbf{v}_{i'j'}^T) = 2(\mathbf{Y}_{i'i'} + \mathbf{Y}_{j'j'}) = 4$.

Proof of Proposition 2. Let $\mathbf{P}' = -\sum_{i'j'} P_{i'j'} v_{i'j'} v_{i'j'}^T$. We define δ_{ij} as

$$\delta_{ij} = \mathbf{e}_i^T \mathbf{e}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

The (k,l) component of the matrix $\mathbf{P}'$ is given by

$$\begin{aligned} \mathbf{P}'_{k,l} &= \mathbf{e}_k^T \left(\textstyle\sum_{i'j'} -\mathbf{P}_{i'j'} \mathbf{v}_{i'j'} \mathbf{v}_{i'j'}^T \right) \mathbf{e}_l \\ &= -\textstyle\sum_{i'j'} P_{i',j'} (\delta_{ki'}\delta_{i'l} + \delta_{kj'}\delta_{j'l} - \delta_{ki'}\delta_{j'l} - \delta_{kj'}\delta_{i'l}) \\ &= P_{k,l}, \text{if } k \neq l \end{aligned} \quad (24)$$

Therefore, the off-diagonal elements of $\mathbf{P}$ and $\mathbf{P}'$ are equal. Since the diagonal elements of $\mathbf{P}$ are all 0 and therefore, not considered in equation (10), we get $\mathbf{P} \bullet \mathbf{Y} = \mathbf{P}' \bullet \mathbf{Y} = -\sum_{i'j'} \mathbf{P}_{i'j'} \mathbf{v}_{i'j'} \mathbf{v}_{i'j'}^T \bullet \mathbf{Y} = -\sum_{i'j'} \mathbf{P}_{i'j'} \mathbf{v}_{i'j'}^T \mathbf{Y} \mathbf{v}_{i'j'} = -\sum_{i'j'} \mathbf{P}_{i'j'} z_{i'j'}$.

Proof of Proposition 3. Consider the first triangular inequality. We get

$$\begin{aligned} z_{ij} + z_{jk} + z_{ik} &= \mathbf{v}_{ij}^T \mathbf{Y} \mathbf{v}_{ij} + \mathbf{v}_{jk}^T \mathbf{Y} \mathbf{v}_{jk} + \mathbf{v}_{ik}^T \mathbf{Y} \mathbf{v}_{ik} \\ &= 2(\mathbf{Y}_{ii} + \mathbf{Y}_{jj} + \mathbf{Y}_{kk}) - 2(\mathbf{Y}_{ij} + \mathbf{Y}_{jk} + \mathbf{Y}_{ik}) \\ &= 6 - 2(\mathbf{Y}_{ij} + \mathbf{Y}_{jk} + \mathbf{Y}_{ik}). \end{aligned} \quad (25)$$

Since $(\mathbf{Y}_{ij} + \mathbf{Y}_{jk} + \mathbf{Y}_{ik}) \geq -1$, it follows that $z_{ij} + z_{jk} + z_{ik} \leq 8$. The other inequalities can be proved similarly.

Acknowledgments This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

[1] http://www.mosek.com.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001.

[3] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *CVIU*, 89(2-3):114–141, 2003.

[4] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *CVPR*, pages I: 10–17, 2005.

[5] F. Kahl. Multiple view geometry and the l-infinity norm. In *ICCV*, 2005.

[6] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *PAMI*, 26(2):147–159, 2004.

[7] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered pictorial structures from video. In *ICVGIP*, pages 148–153, 2004.

[8] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *CVPR*, pages 18–25, 2005.

[9] M. Lobo, L. Vandenberghe, L. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

[10] T. Meltzer, C. Yanover, and Y. Weiss. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *ICCV*, pages I: 428–435, 2005.

[11] M. Muramatsu and T. Suzuki. A new second-order cone programming relaxation for max-cut problems. Technical Report CS-01-01, UEC, 2001.

[12] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *UAI*, 1999.

[13] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kauffman, 1998.

[14] C. Schellewald and C. Schnorr. Subgraph matching with semidefinite programming. In *IWCIA*, 2003.

[15] P. H. S. Torr. Solving markov random fields using semidefinite programming. In *AISTATS*, 2003.

[16] M. Wainwright, T. Jaakkola, and A. Willsky. MAP estimation via agreement on (hyper)trees. Technical Report UCB/CSD-03-1226, UC Berkeley, 2003.

[17] J. Yedidia, W. Freeman, and Y. Weiss. Bethe free energy, kikuchi approximations, and belief propagation algorithms. Technical Report TR2001-16, MERL, 2001.