A Global Perspective on MAP Inference for Low-Level Vision

Oliver J. Woodford*
Department of Engineering Science
University of Oxford

Carsten Rother*
Microsoft Research
Cambridge, UK

Vladimir Kolmogorov*
Department of Computer Science
University College London, UK

Abstract

In recent years the Markov Random Field (MRF) has become the de facto probabilistic model for low-level vision applications. However, in a maximum a posteriori (MAP) framework, MRFs inherently encourage delta function marginal statistics. By contrast, many low-level vision problems have heavy tailed marginal statistics, making the MRF model unsuitable. In this paper we introduce a more general Marginal Probability Field (MPF), of which the MRF is a special, linear case, and show that convex energy MPFs can be used to encourage arbitrary marginal statistics. We introduce a flexible, extensible framework for effectively optimizing the resulting NP-hard MAP problem, based around dual-decomposition and a modified min-cost flow algorithm, and which achieves global optimality in some instances. We use a range of applications, including image denoising and texture synthesis, to demonstrate the benefits of this class of MPF over MRFs.

1. Introduction

A standard approach to solving vision problems today is to specify a probability distribution over the space of output solutions, then (try to) find that solution with the highest probability—the *maximum a posteriori* (MAP) solution when a prior model of the likelihood of output solutions is incorporated using Bayes' rule. Priors generally model dependencies between the output variables; often these dependencies are local, between each output variable and a small number of its closest neighbours, generating a Markov Random Field (MRF). Indeed, the vast majority of low-level vision problems, examples of which include image segmentation, stereo, optical flow and image denoising, employ this framework. As a result, much effort has been invested over the past decade or so into improving optimization of this class of problem, to the extent that other forms of model are currently at a disadvantage in this respect.

However, MRF[1] prior models suffer from a major drawback: the marginal statistics[2] of the most likely solution under the model generally do not match the marginal statistics used to create the model. For example, given a corpus of binary training images which each contain 55% white and 45% black pixels (with no other significant statistic), a learned MRF prior will give each output pixel an independent probability of 0.55 of being white. Since the most likely value for each pixel is white, the most likely image under the model has 100% white pixels, which compares unfavourably with the input statistic of only 55%. When combined with data likelihoods, this model will therefore incorrectly bias the MAP solution towards being all white, the more so the greater the noise and hence data uncertainty. This observation can be extended to any MRF whose marginal statistics are not delta functions[3].

This bias away from the true marginal statistics towards a delta distribution will not be a problem if either the data likelihoods are sufficiently strong so as to make the bias negligible, or the marginal statistics are of secondary importance. The former is a function of the data, while the latter is a function of the application. However, a large number of low-level vision applications rely heavily on the importance of marginal statistics. Image denoising is a classic example—marginal distributions of zero-mean filter responses are typically highly kurtotic (heavy tailed), a statistic often cited as significant for the purpose of denoising, but the MRFs typically used encourage the output statistics to be less kurtotic. Clearly, finding the MAP solution with an MRF prior model is not suitable in this situation.

One way round this problem is to change the method of inference from that of finding the MAP solution to a sampling-based approach [5]. Their prior models are learned by maximizing the likelihood of the training data. Hence, it can be expected that *on average* a random sample will match the marginal statistics quite well, see *e.g.* [20]. However, we are not aware of any sampling technique which attempts to have a *single* output labelling matching the given statistics, which is the goal of this work. There are many other interesting differences, like the choice of loss function, which are, however, not directly relevant for this paper.

If one is to stick with an MAP framework but avoid the bias problem then it is the prior model that must be changed. In particular, a necessary further constraint on the model is that the marginal statistics of the most likely output(s) under

*The authors contributed equally to this work, therefore assert joint first authorship.

[1] We assume that MRFs are translationally *invariant*. We refer to a translationally *variant* MRF as a Conditional Random Field (CRF).

[2] We refer specifically to the marginal statistics of the cliques used in the model, which generally equates to those statistics deemed important.

[3] In this case the most likely output under the model will also have delta function marginal statistics.

the model match the marginal statistics of the training data.

The way to achieve the latter is to have a prior over the marginal statistics of the output solution. Since computing marginal statistics involves every output variable, this model generates a single clique over all variables—what we call a Marginal Probability Field (MPF)—which means that the powerful optimization techniques developed for solving MRFs, with their small cliques, are not suitable. In fact, there is a paucity of efficient optimization techniques for MPFs, which we believe has slowed their adoption in comparison to the inferior MRF model. The goal of this paper is redress the balance by developing a powerful inference framework for optimizing the MAP problem that results from an MPF.

Related work The task of having the output labelling match a given distribution has been addressed in different ways in the literature. There are many vision systems which have a global prior built into their models, *e.g*. [27, 20, 21]. However, these works use sampling to produce an output, which has a different goal in mind than our approach, as discussed above. For instance, Sudderth *et al*. [20] recently introduced the Pitman-Yor process to match the heavy-tailed distribution of object labels, in an object recognition and segmentation framework.

In the context of MAP inference there are only a few papers which address the bias of the prior. Most of them tackle the problem by building new approximate inference methods in order to match the target statistics, without any global optimality guarantees. For texture synthesis, both [7] and [26] compute second-order marginal statistics of an input texture over a range of different pairwise neighbourhood structures. While [7] used ICM, [26] enforced the target statistics by adapting a Metropolis sampling procedure. For the same problem, Kopf *et al*. [14] enforce a global first-order statistic (*i.e*. unary cliques) on colour. They used an approximate EM-style algorithm for MRF optimization, which we compare to in section 4.4. Other related works address the problem of binary segmentation where the appearance of the foreground segment has to match a given distribution [17, 11, 6]. While [6] used active contours, [17] developed a trust region graph cut approach, which we will compare in our experiments.

In this paper we use the *dual decomposition approach* [23, 18, 19, 13, 24]. In contrast to approximate methods, it provides a lower bound which can be used to achieve global optimality, as we will see for some cases. We introduce two new methods for use within this framework. The first one enforces the area constraint of binary segmentations defined on a tree; this often gives a tighter lower bound compared to the one in [24]. Our second method is based on a modified min-cost flow algorithm; it is able to handle convex terms of global statistics with an arbitrary number of labels. A more detailed discussion and further related work is presented in sec. 3.

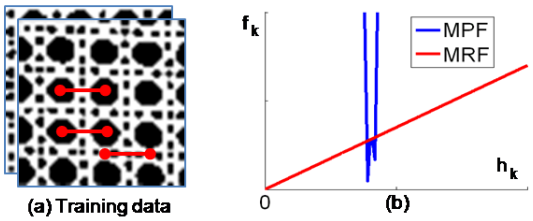


Figure 1. **MPF versus MRF**. (a) Set of training images for binary texture denoising. Superimposed is a pairwise feature (translationally invariant pairwise terms with shift $(15, 0)$; 3 exemplars in red). Each pairwise feature has one histogram value h_k per training image (with $k \in \mathcal{K}$, and $|\mathcal{K}| = 4$). (b) The trained MPF cost kernel $f_k\,(h_k)$, *i.e*. the negative log of probability of h_k over training images (here $k = (1, 1)$). The occurrence of label $(1, 1)^T$ is nearly the same for all training images (of same size). It is apparent that the linear cost function of an MRF is a bad fit.

2. Marginal probability fields (MPFs)

When creating a prior probability model, one typically chooses[4] a subset of features which depend on the output $\boldsymbol{x}$. For example, in stereo this might be the derivative of disparity (uni-dimensional), while in texture synthesis it might be a 5×5 image patch (multi-dimensional). These features are computed over neighbourhoods of the output solution.

In an MRF, these features define an **independent** cost for every such neighbourhood in $\boldsymbol{x}$. Its energy[5] is written as

$$E_{\text{MRF}}(\boldsymbol{x}) = \sum_i f_{\text{MRF}}\left(\phi_i(\boldsymbol{x})\right), \tag{1}$$

where $\phi_i : \mathbb{R}^{|\boldsymbol{x}|} \rightarrow \mathbb{R}^n$ computes the n-d feature vector centred on element i of $\boldsymbol{x}$ and $f_{\text{MRF}} : \mathbb{R}^n \rightarrow \mathbb{R}^+$ is the clique cost functional. For example, in binary texture denoising (see details in sec. 4.2), $\boldsymbol{x}_i \in \{0, 1\}$, and a pairwise feature is of the form $\phi_i(\boldsymbol{x}) \in \{(0, 0)^T, (0, 1)^T, (1, 0)^T, (1, 1)^T\}$ (see *e.g*. fig. 1(a)).

By contrast, in an MPF the likelihoods of these features are **not independent**. Rather, probability is computed as a function of the marginal statistics of the features. The energy of an MPF is therefore written as

$$E_{\text{MPF}}(\boldsymbol{x}) = \int f_k \left(\sum_i \left[\phi_i(\boldsymbol{x}) = k\right] \right) \mathrm{d}k, \tag{2}$$

where k is an n-d feature vector, f_k is the MPF cost kernel $\mathbb{R} \rightarrow \mathbb{R}^+$ for a given feature vector and $[\cdot]$ is the Iverson bracket[6]. Normally the feature vector space is discretized into a set of labels, $\mathcal{K}$, allowing a histogram, $\{h_k\}_{k \in \mathcal{K}}$, to be computed as $h_k = \sum_i [\phi_i = k]$ (where ϕ_i replaces $\phi_i(\boldsymbol{x})$

[4] Even with learned features, there is a decision about what class of features they should be learned from.

[5] Energy is the negative log of (usually unnormalized) probability.

[6] [statement] $= 1$ if statement is true, 0 otherwise.

for brevity), such that $E_{\text{MPF}}(\boldsymbol{x}) = \sum_{k\in\mathcal{K}} f_k\left(h_k\right)$. Fig. 1 illustrates a learned function f_k for the denoising example.

2.1. Characterizing cost functions

The cost function f_k has a large bearing on the usefulness of an MPF, and also the optimizability of the resulting energy minimization problem. We categorize such functions according to their second derivative w.r.t. frequency,

$$f_k''(h_k) = \frac{\partial^2 f_k(h_k)}{\partial h_k^2}, \tag{3}$$

and discuss the merits of three main classes.

Linear MPFs have $f_k'' = 0,\ \forall\ k, h_k$, with the result that $f_k(h_k) = \mathrm{c}_k + f_k \cdot h_k$. Such an MPF is equivalent[7] to an MRF:

$$E_{\text{MPF}}(\boldsymbol{x}) = \sum_k f_k \cdot \sum_i \left[\phi_i = k\right] = \sum_i \sum_k f_k \cdot \left[\phi_i = k\right] = \sum_i f_{\text{MRF}}\left(\phi_i\right) \ = \ E_{\text{MRF}}(\boldsymbol{x})$$

therefore can be optimized using standard MRF optimizers.

Concave MPFs have $f_k'' \leq 0,\ \forall\ k, h_k$. The global minimum of any MPF in this (and therefore also the linear) class will generally, ignoring integrability[8], produce delta function marginal statistics. This can be seen from the fact that a concave function $\sum_k f_k(h_k)$ defined over a simplex $\{\{h_k\} : h_k \geq 0, \sum_k h_k = const\}$ attains a minimum at an extreme point of this simplex.

Convex MPFs have $f_k'' \geq 0,\ \forall\ k, h_k$, and can generate *arbitrary* marginal statistics, as we will show. This paper introduces a powerful optimization framework for this class of problem which produces good results and in some cases even finds a global optimum. Cost functions may of course have both regions of convexity and concavity, but the properties of such functions are not discussed here. In fig. 1(b) we see that a convex function would be a good fit to the learned function f_k.

2.2. Specifying cost functions

When defining the parameters of an MPF one either has a single marginal statistic, *i.e.* histogram $\{\bar{h}_k\}_k$, for each feature, or a training set of such marginal statistics, $\{\{\bar{h}_k^d\}_k\}_d$. The first case occurs when there is a single training image, such as in image synthesis from a single exemplar image, or a user-defined defined statistic, for example the area or colour histogram of an object to be segmented or tracked. In this case it is clear that each cost function, f_k, should be unimodal and that its minimum should occur at $\bar{h}_k$, to encourage the output to have the same marginal statistics. However, the *shape* of the unimodal cost function is an open variable. Experimentation or learning with ground truth output is a future possibility, but in this work we generally use the "V-shaped" kernel.

In the second case the training data provides a set of histograms. Example applications are image and binary texture denoising, where a set of training images are available.

In this case the training data can be used to specify the shape of the cost functions, *e.g.* by histogramming the set $\{\bar{h}_k^d\}_d$ for each k (see fig. 1(b)). In this work we are limited to convex, piecewise-linear kernels. It is not yet clear for which applications these statistics tend to be convex, but we show that simple "V-shaped" kernels can achieve good results for image denoising and binary texture denoising.

The simplicity of our cost functions stems from the fact that the main purpose of this paper is not parameter learning, but rather to show that the convex energy MPF is a better prior model than an MRF for most low-level vision applications using MAP inference. This will hopefully inspire further research on the subject of learning MPF parameters.

2.3. Incorporating MRFs into MPFs

Posterior probability models generally contain a number of different terms, making the total energy a sum of, for example, data likelihoods, MRFs and/or CRFs over various different features; let f_{MRF} encompass such costs. We can incorporate these costs into the MPF, and also extend the MPF over various different features, redefining E_{MPF} thus[9]

$$E_{\text{MPF}}(\boldsymbol{x}) = f_{\text{MRF}}(\boldsymbol{x}) + \sum_t \sum_k f_k^t(h_k^t)\,. \tag{4}$$

Here we assume that there are several types of features indexed by symbol t. h_k^t denotes the histogram of label k over features of type t: $h_k^t = \sum_i [\phi_i^t = k]$ where $\phi_i^t = \phi_i^t(\boldsymbol{x})$ is the feature of type t at location i taking values in some finite set $\mathcal{K}^t$. In binary texture denoising $\mathcal{K}^0 = \{0, 1\}$ and $\mathcal{K}^1 = \{(0,0)^T, (0,1)^T, (1,0)^T, (1,1)^T\}$ for the unary and pairwise terms respectively. In our experiments we will demonstrate various types of MPFs based on unary and pairwise terms, and leave higher-order clique MPFs as future work.

3. Optimization

We now concentrate on the problem of optimizing MPF energies given by (4). In general, this is a very challenging task. We are aware of only a few special cases that can be solved exactly in polynomial time. A notable example is given in [9]: if $\boldsymbol{x}$ is a binary labelling, $f_{\text{MRF}}(\boldsymbol{x})$ is a submodular function with unary and pairwise terms, features ϕ_i^t correspond to individual pixels ($\phi_i^t(\boldsymbol{x}) = x_i$), and functions $f_k^t(\cdot)$ are **concave** then the problem can be solved via

[7] The constants, c_k, are removed for brevity, without loss of generality.

[8] Independently selecting the most likely labelling for every clique is not possible where intersections between clique labellings differ.

[9] Note, eq. (4) can be viewed as a special case of (2), if the domain of features k in (2) is defined appropriately as a Cartesian product. The form of eq. (4), however, will be more convenient in sec. 3.

a reduction to a min s-t cut problem. In this paper, however, we are more interested in the case of **convex** functions $f_k^t(\cdot)$ which is well-known to be NP-hard (since *e.g.* it includes the *minimum graph bisection* problem as a special case).

As discussed in section 1, many heuristic techniques have been proposed. This section, however, considers only global methods that provide a lower bound on the energy. In particular, we will use the *subproblem decomposition* (or *dual decomposition* - DD) approach [3], which proved to be very successful for MRF optimization [23, 18, 19, 13]. Note that such an approach was used in [24] for enforcing a statistic on the area of a binary segmentation.

Let us introduce vector $\boldsymbol{\theta} = \{\theta^t\}_t = \{\theta^t_{ik}\}_{t,i,k}$; we will denote $\theta^t_i(k) = \theta^t_{ik}$. We can rewrite the energy (4) as

$$E_{\text{MPF}}(\boldsymbol{x}) = E_{\text{MRF}}(\boldsymbol{x};\boldsymbol{\theta}) + \sum_t E^t(\boldsymbol{x};\theta^t) \text{ where} \quad (5)$$

$$E_{\text{MRF}}(\boldsymbol{x};\boldsymbol{\theta}) = f_{\text{MRF}}(\boldsymbol{x}) - \sum_t \sum_i \theta^t_i(\phi^t_i) \quad (6)$$

$$E^t(\boldsymbol{x};\theta^t) = \sum_k f^t_k(h^t_k) + \sum_i \theta^t_i(\phi^t_i)\,. \quad (7)$$

In the decomposition approach we define lower bounds for individual terms:

$$\Phi_{\text{MRF}}(\boldsymbol{\theta}) \leq \min_{\boldsymbol{x}} E_{\text{MRF}}(\boldsymbol{x};\boldsymbol{\theta}) \quad (8)$$

$$\Phi^t(\theta^t) \leq \min_{\boldsymbol{x}} E^t(\boldsymbol{x};\theta^t)\,. \quad (9)$$

The sum of these bounds then gives a lower bound on the original function:

$$\Phi(\boldsymbol{\theta}) = \Phi_{\text{MRF}}(\boldsymbol{\theta}) + \sum_t \Phi^t(\theta^t) \leq \min_{\boldsymbol{x}} E_{\text{MPF}}(\boldsymbol{x})\,. \quad (10)$$

In order to get the tightest possible bound on $E_{\text{MPF}}(\boldsymbol{x})$, we need to maximize function $\Phi(\boldsymbol{\theta})$ over $\boldsymbol{\theta}$. Bounds (8) and (9) are chosen in such a way that function $\Phi(\cdot)$ is concave, therefore one can use a number of standard concave maximization techniques. Following [18, 19, 13], we maximize $\Phi(\cdot)$ via a subgradient technique.

Let us now discuss how to define bounds (8) and (9). Possible bounds $\Phi_{\text{MRF}}(\boldsymbol{\theta})$ on MRF functions have been extensively studied before. A popular choice is to use a convex combination of trees [23]; more generally, one can use subproblems with low tree-width. We therefore focus on lower bounds $\Phi^t(\theta^t)$ on global statistics terms, which have received less attention in the literature.

Lower bounds on global statistics terms From now on, we consider a fixed index t. For brevity, we will denote $f_k = f^t_k$, $\theta = \theta^t$, $\mathcal{K} = \mathcal{K}^t$. As we just discussed, our goal is to define a lower bound on function (7):

$$E^t(\boldsymbol{x};\theta) = \sum_k f_k\left(\sum_i [\phi^t_i = k]\right) + \sum_i \theta_i(\phi^t_i)\,. \quad (11)$$

Ideally, we would like to take $\min_{\boldsymbol{x}} E^t(\boldsymbol{x};\theta)$ as the lower bound. Unfortunately, computing this minimum is an NP-hard problem even in rather restricted cases.[10] To get a tractable lower bound, we replace features ϕ^t_i in (11) with labels $k_i \in \mathcal{K}$. We then minimize the energy over labellings $\mathbf{k}$ **without** enforcing the constraint that $\mathbf{k} = \phi^t(\boldsymbol{x})$ for some labelling $\boldsymbol{x}$. (An example, assume $\phi^t \in \{(0,0),(0,1),(1,0),(1,1)\}$, and consider three nodes x_{1-3} where $\phi^t(x_1,x_2) = (0,1)$ and $\phi^t(x_2,x_3) = (0,1)$. This provides a valid labelling but an invalid assignment for x_2.) Thus,

$$\Phi^t(\theta) = \min_{\mathbf{k}} \tilde{E}^t(\mathbf{k};\theta) \text{ where} \quad (12)$$

$$\tilde{E}^t(\mathbf{k};\theta) = \sum_k f_k(h_k(\mathbf{k})) + \sum_i \theta_i(k_i) \quad (13)$$

$$h_k(\mathbf{k}) = \sum_i [k_i = k] \quad (14)$$

In the remainder of this section we discuss how to solve the minimization problem (12). We will consider separately the case of binary variables k_i ($|\mathcal{K}| = 2$) and multi-valued variables ($|\mathcal{K}| > 2$).

3.1. Case I: binary variables

In this section we assume that $k_i \in \mathcal{K} = \{0,1\}$ and θ^t_{i0} is constrained to be 0 for all locations i (possible since adding a constant to θ^t_{i0} and θ^t_{i1} does not change $\Phi(\boldsymbol{\theta})$ for bounds $\Phi_{\text{MRF}}(\boldsymbol{\theta})$ used in the literature). We can rewrite (13) as

$$f(h_1(\mathbf{k})) + \sum_i \theta_i k_i \quad (15)$$

where $f(h_1) = f_1(h_1) + f_0(n - h_1)$, $n = \sum_i 1$ is the total number of elements i, and $\theta_i = \theta_{i1}$. (We used the fact that $h_0(\mathbf{k}) = n - h_1(\mathbf{k})$.)

It is well-known that the minimum of (15) can be computed in $O(n \log n)$ time [8]. We need to sort values θ_i in non-decreasing order, evaluate the cost of $n+1$ labellings $(0,\ldots,0)$, $(1,0,\ldots,0)$, $(1,1,0,\ldots,0)$, …, $(1,\ldots,1)$ (where we assume that the order of elements is given by the sorting), and pick the labelling with the smallest cost. A decomposition with subproblem 15 was used as an example in [24] for enforcing the area constraint of a binary segmentation.

Convex case Suppose that function $f(\cdot)$ is convex. This case is quite special due to the following theorem proved in [22].

[10] Suppose, for example, that $\boldsymbol{x}$ is a binary labelling ($x_p \in \{0,1\}$), i indexes an edge $(p(i),q(i))$, feature $\phi^t_i = |x_{p(i)} - x_{q(i)}| \in \{0,1\}$ measures the discontinuity between nodes $p(i)$ and $q(i)$, term $f_0(\cdot)$ is a linear function: $f_0(h) = h$, and other terms in (11) are identically zero: $f_1(h) = 0$, $\theta_i(k) = 0$. Then $E^t(\boldsymbol{x};\theta)$ equals the number of edges $(p(i),q(i))$ with the same label ($x_{p(i)} = x_{q(i)}$). Minimizing $E^t(\boldsymbol{x};\theta)$ is thus equivalent to a maximum cut problem in an undirected unweighted graph, which is well-known to be NP-hard.

Theorem 3.1 *Suppose that function $\Phi_{MRF}(\cdot)$ is continuous and satisfies the following property for all vectors $\boldsymbol{\theta}$ and locations i of feature of type t:*

$$\Phi_{MRF}(\boldsymbol{\theta} + \delta \cdot \chi_i) \geq \Phi_{MRF}(\boldsymbol{\theta}) + \min_{x \in \{0,1\}} \{-x\delta\} \qquad (16)$$

where χ_i is a vector of the same dimensions as $\boldsymbol{\theta}$ with $(\chi_i)^t_{i1} = 1$ and all other components equal to 0. Then function (10) has a maximizer $\boldsymbol{\theta}$ such that $\theta^t_{i1} = \lambda$ for some constant λ.

Note, condition (16) is satisfied for many reasonable choices of $\Phi_{\text{MRF}}(\cdot)$; in particular it holds if $\Phi_{\text{MRF}}(\cdot)$ is defined to be equal to the minimum of (6).

The theorem allows us to restrict vector θ^t to have the form $\theta^t = \lambda \mathbb{1}$ where $\mathbb{1}$ is a vector of size $2n$ with $\mathbb{1}_{i0} = 0$ and $\mathbb{1}_{i1} = 1$. This should speed up a subgradient method. Furthermore, if function (4) has no other global terms except for $f(h_1(\phi^t))$ then the bound $\Phi(\cdot)$ depends just on a single parameter λ, hence one can use *e.g.* a line search. Evaluating $\Phi^t(\lambda\mathbb{1})$ is straightforward, so the bottleneck computation is evaluating $\Phi_{\text{MRF}}(\lambda\mathbb{1})$ for different λ's.

In the experiments we consider the problem of minimizing a submodular function with unary and pairwise terms plus a global convex term of the area of the binary segmentation. In this case $\Phi_{\text{MRF}}(\lambda\mathbb{1}) = \min_{\boldsymbol{x}}\{\Phi_{\text{MRF}}(\boldsymbol{x}) - \lambda \sum_i x_i\}$ where $\Phi_{\text{MRF}}(\cdot)$ is a submodular function with unary and pairwise terms. It is well-known that the minimum can be computed efficiently for all values of λ using a parametric maxflow algorithm, see *e.g.* [11]. We denote this method as **DD**.

3.1.1 Adding pairwise terms

For certain problems the bound defined by (10) can be quite loose. We now discuss one possible way to improve it. Let us add to function (15) pairwise terms defined on a tree (or a forest) T:

$$f(h_1(\mathbf{k})) + \sum_i \theta_i k_i + \sum_{(i,j)\in T} f_{ij}(k_i, k_j)\,. \qquad (17)$$

Clearly, the minimum of (17) can be computed in $O(n^2)$ time using dynamic programming (see [25] for details). In our experiments we used it for minimizing function $E(\boldsymbol{x}) = f(\sum_p x_p) + \sum_p f_p(x_p) + \sum_{(p,q)\in\mathcal{E}} f_{pq}(x_p, x_q)$ of binary labellings $\boldsymbol{x}$ as follows. First, we divide the edges into T disjoint groups ($\mathcal{E} = \cup_t \mathcal{E}^t$) so that each group $\mathcal{E}^t$ forms a forest. We then use the approach described at the beginning of section 3, only instead of (5) we use the decomposition $E(\boldsymbol{x}) = \sum_t [\frac{1}{T} f(\sum_p x_p) + \frac{1}{T}\sum_p f_p(x_p) + \sum_{(p,q)\in\mathcal{E}_t} f_{pq}(x_p, x_q) + \sum_p \theta^t_p x_p]$ where θ^t are unknown vectors that must sum to 0. Each term in this decomposition is minimized over $\boldsymbol{x}$ via dynamic programming. We denote this method as **DD-DP**.

3.2. Case II: multi-valued variables

Let us now consider the case of multi-valued variables ($K = |\mathcal{K}^t| > 2$). This case was recently analyzed in [8]. The authors mention that the problem can be solved in $O(n^K)$ time, which unfortunately would be too slow in practice even for small K. The work [8] focuses on the case when $f_k(h_k(\mathbf{k})) = \lambda(h_k(\mathbf{k}))^2$. The authors observed that if $\lambda > 0$ then the problem can be solved via quadratic programming. They then proposed an approximation algorithm for the case $\lambda < 0$, which they proved to be NP-hard.

Below we present further results for the case when $f_k(\cdot)$ are arbitrary **convex** functions. We show that the minimization problem (12) can then be reduced to a *minimum cost flow* problem (MCF) in a bipartite graph. If all input costs are integers then one could apply the MCF algorithm for bipartite graphs in [2]; the complexity would be $O(nK^2 + K^3 \log(KC))$ where $n = \sum_i 1$ is the number elements i and C is the largest cost. In this paper we used an alternative algorithm with a strongly polynomial complexity $O(nK^3 \log(n + K))$, which we developed.

We denote the dual decomposition approach with this method as **DD-MCF**.

Reduction to MCF We assume that the reader is familiar with the MCF problem (see *e.g.* [1] for details). We construct a graph with n nodes corresponding to elements i (called *i-nodes*) and K nodes corresponding to labels (called *k-nodes*). We also add one extra node s called the *source* (fig. 2). For each label k and element i we add an arc $i \to k$ with capacity 1 and cost θ_{ik}. Each i-node will have an excess flow of +1. This unit of flow must leave i via some arc $i \to k$; this will correspond to assigning label k to element i: $k_i := k$.

Now consider label k. The argument of $f_k(\cdot)$ can only take values $0, 1, \ldots, n$, therefore we can assume without loss of generality that $f_k(\cdot)$ is a piecewise-linear convex function that attains a minimum of 0 in $[0, n]$. Let $0 \leq n_1 < \ldots < n_B \leq n$ be the breakpoints of $f_k(\cdot)$ and let $s_0 < s_1 < \ldots < s_B$ be the corresponding slopes of linear segments, with $s_0 \leq 0$ and $s_B \geq 0$. Let $n_{\bar{b}}$ be a breakpoint that minimizes $f_k(\cdot)$, then $s_{\bar{b}-1} \leq 0$ and $s_{\bar{b}} \geq 0$. We set the excess (or rather the deficit) of node k to $-n_{\bar{b}}$. For $b = 1, \ldots, \bar{b}$ we add an arc from s to k with capacity $n_b - n_{b-1}$ and cost $-s_{b-1}$. For $b = \bar{b}, \ldots, B$ we add an arc from k to s with capacity $n_{b+1} - n_b$ and cost s_b.

To complete the construction, we assign an excess/deficit to the source node to make the network balanced. (Recall that in the MCF formulation the sum of excesses over all nodes should be zero.)

It is not difficult to see that the cost of a valid flow equals the cost (13) for the corresponding labelling $\mathbf{k}$. Indeed, arcs from i-nodes to k-nodes incur cost $\sum_i \theta_i(k_i)$. Now consider node k. The amount of flow that comes to k from

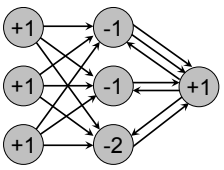


Figure 2. **Reduction to min-cost flow.** Example for a function with three pixels and three labels ($n = K = 3$). Term $f_1(\cdot)$ has two breakpoints, terms $f_2(\cdot), f_3(\cdot)$ have one breakpoints.

i-nodes is $h_k(\mathbf{k})$. If it is equal to the deficit of $n_{\bar{b}}$ at node k then no flow will go between i and s, so edges between k and s will not incur any cost. If $h_k(\mathbf{k})$ exceeds $n_{\bar{b}}$ then the additional flow will leave k via arcs $k \rightarrow s$, incurring the cost $f_k(h_k(\mathbf{k}))$. Similarly, if $h_k(\mathbf{k})$ is less than $n_{\bar{b}}$ then some flow will go from s to k to cancel the deficit at k.

Solving the MCF problem The constructed network contains $n + K + 1$ nodes and $O(nK)$ arcs. A general-purpose MCF solver applied to this problem would be quite slow; for example, the successive shortest path (SSP) algorithm would have $O(nK)$ iterations consisting of Dijkstra computations, resulting in $O(n^2K^2 \log(n + K))$ time. We developed a modification of the SSP algorithm that runs in $O(nK^3 \log(n + K))$; details are given in [25]. After the submission we learned about an alternative MCF algorithm for bipartite graphs [2] which would have $O(nK^2 + K^3 \log(KC))$ complexity for integer costs.[11] Unlike our algorithm it is not strongly polynomial since it depends on the largest cost C, but we expect it to be faster in practice.

4. Experimental results

Our experiments concern two questions. Firstly, how well can the dual-decomposition approach optimize the MPF model, especially w.r.t. competitive methods. Secondly, how does the MPF model compare to a standard MRF model. For this, we have considered four different applications: image segmentation, synthesis and denoising, and binary texture denoising. Note, further results are in [25].

Figure 3. **Image segmentation** with standard MRF [16] (*centre*) and MPF with area (global unary) constraint (*right*).

4.1. Image Segmentation

We used the GrabCut MRF model with the provided dataset of 50 images [16].[12] Fig 3(centre) shows a result, where colours were trained from the provided user-defined trimap [16]. In order to improve on this result we used two types of high-level knowledge (*e.g.* from user or previous frame in tracking). Firstly, we used a global constraint

[11] We thank Andrew Goldberg for pointing out this reference.

[12] We downscaled images to max side length of 70 pixels, which only mildly affects segmentation quality, in order to run many experiments.

	MRF	MPF - global area		MPF - global distribution	
		DD	DD-DP	DD	TRGC
hard	2.8	2.6 (11)	2.6 (37)	2.1 (4.3)	2.5
soft	2.8	2.4 (39)	2.5 (57)	1.9 (28)	2.0

Table 1. **Image segmentation** with different models and methods. Shown is error (percentage of misclassified pixels) and in brackets percentage of globally optimal cases (TRGC has no guarantees).

Noise	MRF	MPF - global unary		MPF - global pair.
		DD	DD-DP	DD-MCF
30%	6.9	6.1 (57)	6.2 (50)	6.3 (0)
60%	20.1	14 (23)	13.7 (11)	12.5 (0)
90%	40.6	36.8 (15)	33.4 (0)	31.3 (0)

Table 2. **Texture denoising** with different models, methods and noise levels. Shown is error (percentage of misclassified pixels) and in brackets percentage of global optimal cases.

on the foreground area (defined by the ground truth segmentation), enforced either as soft (with V-shaped cost kernel) or hard constraint, *i.e.* foreground area perfectly matching ground truth. Table 1 shows results averaged over the dataset. As expected, the error rate with the area constraint (MPF - global area) is lower than without (MRF); example in fig. 3(*right*).[13] More importantly, we obtain global optimality for many examples; in particular, this is achieved more often when using DD-DP (sec. 3.1.1) rather than DD (sec. 3.1), and also when using a soft constraint. In a second experiment, we converted the independent, unary label costs of the MRF (based on foreground and background colour models) into an MPF with fixed-size target colour histograms for foreground and background regions and V-shaped cost kernels over each colour bin,[14] as per [17]. This is a stronger constraint than global area, with error rates improving considerably (see table 1, MPF - global distribution). Our DD method is globally optimal for some cases, and gives on average (94% "hard" case, 72% "soft" case) a lower energy than the approximate trust region graph cut (TRGC) method.[15]

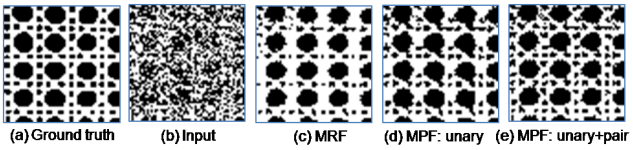


Figure 4. **Binary texture denoising** with various models (crop 60×60 pixels of Brodatz D101). The input image (b) has 60% noise. The error rates are (c) 13.7%, (d) 12.8%, (e) 10.6%.

4.2. Binary texture denoising

This toy problem (see fig. 4, 1) has been addressed before using pairwise MRFs [12, 4]—using a set of (here,

[13] The slightly higher error for hard constraints is due to the fact that the ground truth segmentation is not perfectly aligned with edges in an image.

[14] The two colour histograms are transformed so that their n colour bins become the features, and the labels foreground and background become the bins, creating n binary, global subproblems.

[15] We selected the best performing method from [11], *i.e.* strategy C.

80) training images, the 6 most "informative" pairwise features (edges with different length and orientation) are selected (see [4]), their edge costs defined by the negative log of the marginal probabilities of each of the labels $\{(0,0)^T,(0,1)^T,(1,0)^T,(1,1)^T\}$, and the prior weight learned discriminatively (see [12]). The globally optimal labelling of the resulting MRF (found using QPBO [12]) is over-smoothed (fig. 4(c)), due to the bias towards delta function marginal statistics.

We considered two alternatives for improving results. Firstly, we added a V-shaped global unary constraint on the number of 1s. Secondly, we replaced the MRF pairwise terms with V-shaped global potentials (and also kept the global unary constraint). In both cases we defined the lowest cost to be at the mean frequency of the training data statistics, with the kernel gradients being hand-tuned![16] Results, shown in fig. 4(d,e), are superior to the MRF result.

Table 2 provides some quantitative results, here averaged over 20 sample runs (for each noise level) and two different crops (60×60 pixels) of Brodatz textures D101 and D20, reinforcing that the MPF models improve over the MRF. As expected, the difference is more noticeable for higher levels of noise, where the prior has greater influence. The improvement from using global pairwise terms is visually more noticeable (*e.g.* fig. 4(e)) than is reflected by the error rates. In contrast to the previous experiment on image segmentation, DD-DP achieved global optimality less often compared to DD. In particular, we observed that they achieve global optimality for different images. Also, the lower bound of DD-DP was seldom higher than that of DD, though it nearly always had a lower energy. We conjecture that the main reason might be slow convergence of DD-DP.

The average runtime (3.6GHz) for an example is DD 0.5s, DD-DP 1s per iteration, and DD-MCF 0.06s per iteration. The number of iterations (until process seems to be converged) depends on the image, *e.g.* 400 for DD-DP (fig. 4(d)) and 300 for DD-MCF (fig. 4(e)).

4.3. Image denoising

Responses of natural images to zero-mean filters are known to be highly kurtotic. We use a first derivative filter (horizontal and vertical) as a feature to regularize noisy greyscale images, comparing MPF and MRF models. The image is discretized to 64 grey levels, while the derivative statistics (magnitude only) are discretized into 11 non-uniformly distributed bins. A mean image histogram, and histogram variance, is computed from 100 Berkeley Segmentation Dataset images, and used with a V-shaped cost kernel (the gradient of which is inversely proportional to variance) in the MPF, while its negative log is used as the pairwise energy in the MRF problem. A reasonable prior weight for each model was hand-picked from a range of values tested, though the MPF results were found to be much less sensitive to this value. Fig. 5 shows the results using the two different models. The MRF model (optimized using [10]) biases the output (c,d) towards an unnaturally homogenous image, while the MPF output (e), generated using DD-MCF, is both more natural-looking (also matching the mean marginal statistics very closely (f)) and a more faithful reconstruction (visually) of the original.

4.4. Image synthesis

The goal of image[17] synthesis is to generate, from a small (here, 128×128, *e.g.* fig. 6(a)) exemplar image, a larger output image. The popular MRF-based technique of Kwatra *et al.* [15] fuses shifted copies (we use 70 random shifts) of the input image together in a series of binary optimizations (we use QPBO [12]), by minimizing the pairwise transition costs between different copies. The result (fig. 6(b)) is poor—some rarer elements are lacking, *e.g.* dark grass, cow. This can be expected since elements which occur frequently are more easily pasted together.

We augment the MRF energy with a global term based on V-shaped cost kernel around the colour histogram of the input image. This colour histogram has 32 bins whose centres are computed using *k*-means on the input colours; colours are then assigned to the nearest bin centre. Each binary optimization is then computed using our DD-MCF technique. This generates the output shown in fig. 6(c), which shows that the previously missing image elements have been introduced.

Kopf *et al.* [14] first introduced a heuristic to enforce global colour histograms in texture synthesis, inspiring this choice of application here. However, their method![18] generating fig. 6(d), fails to reproduce the full gamut of input colours, a result of the fact that colours, once lost, are not easily reintroduced. Fig. 6(e) shows the colour histograms of images (a–e), indicating that our approach generates the closest match to the ground truth.

5. Conclusions and future work

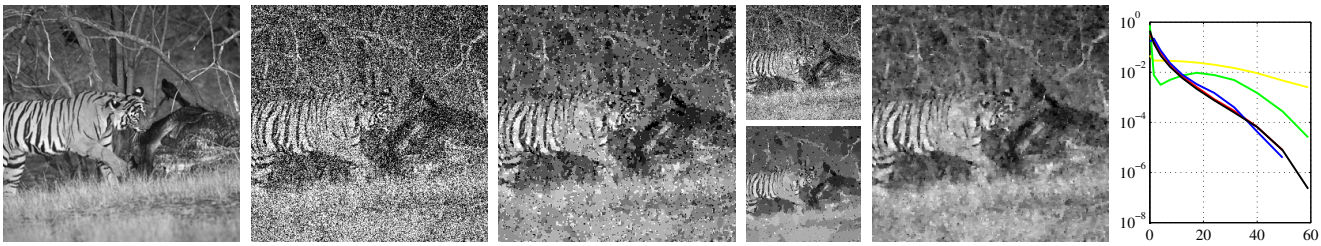
This paper has introduced a framework for the MAP optimization of convex MPFs, powerful in terms of its efficacy, efficiency and flexibility. In doing so it has developed a more general probabilistic model (of which the MRF is a special case), shown that MAP inference *can* generate solutions with correct marginal statistics, when used with a convex MPF, and that convex MPFs generate improved results over those of MRF models in a wide range of low-level vision applications where marginal statistics are both important and heavy-tailed.

We believe that this work has enormous potential, both with respect to the applications it can be applied to, and for

[16] We found that the output labelling was not sensitive w.r.t. settings of the weight, in contrast to a standard MRF.

[17] As distinct from texture—images contain non-repetitive features.

[18] Our own implementation.



(a) Original (b) Noisy input (c) MRF result (d) (e) MPF result (f) Deriv. hists.

Figure 5. **Image denoising**. Results for image denoising. (d) shows further MRF results with a lower (*top*) and higher (*bottom*) prior weight than (c). (f) shows derivative histograms (discretized into the 11 bins used in the global statistic) for the mean statistic, derived from a large dataset (black), (a) (blue), (b) (yellow), (c) (green) and (e) (red). The runtime for the MRF (c) is 1096s and for MPF (e) 2446s.

(a) Exemplar (b) Kwatra *et al*. [15] (c) Our MPF method (d) Kopf *et al*. [14] (e) Colour hists

Figure 6. **Image synthesis**. (a) Input image. (b–d) Larger output images synthesized using (b) the fusion approach of Kwatra *et al*. [15], (c) our adaption which incorporates a global colour histogram constraint, and (d) the EM-style, global heuristic approach of Kopf *et al*. [14]. (e) Colour histograms (bins are those used in the global constraint of (c)) for (a) (black), (b) (red), (c) (blue) and (d) (green).

improvements to the optimization technique itself.

Acknowledgements We thank Tom Minka, Andrew Blake, Pushmeet Kohli, Victor Lempitsky and Andrew Goldberg for helpful discussions.

References

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] R. K. Ahuja, J. B. Orlin, and C. Stein. Improved algorithms for bipartite network flow. *SIAM J. Computing*, 23(5), 1994.

[3] D. Bertsekas. *Nonlinear Programming*. A. Scientific, 99.

[4] D. Cremers and L. Grady. Learning statistical priors for efficient combinatorial optimization via graph cuts. In *ECCV*, 2006.

[5] C. Fox and G. Nicholls. Exact MAP States and Expectations from Perfect Sampling: Greig, Porteous and Seheult revisited. In *AIP conf.*

[6] D. Freedman and T. Zhang. Active contours for tracking distributions. *PAMI*, 13(4), 2004.

[7] A. Gagalowicz and S. D. Ma. Sequential synthesis of natural textures. *Comp. vision, graphics, and image proc.*, 1985.

[8] R. Gupta, A. Diwan, and S. Sarawagi. Efficient inference with cardinality-based clique potentials. In *ICML*, '07.

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

[10] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10), 2006.

[11] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *ICCV*, 2007.

[12] V. Kolmogorov and C. Rother. Minimizing non-submodular functions with graph cuts - a review. *PAMI*, 29(7):1274–1279, 2007.

[13] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007.

[14] J. Kopf, C.-W. Fu, D. Cohen-Or, O. Deussen, D. Lischinski, and T.-T. Wong. Solid Texture Synthesis from 2D Exemplars. *SIGGRAPH*, 26(3):2:1–2:9, 2007.

[15] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *SIGGRAPH*, July 2003.

[16] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, August 2004.

[17] C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into MRFs. In *CVPR*, 06.

[18] M. I. Schlesinger and V. V. Giginyak. Solution to structural recognition (MAX,+)-problems by their equivalent transformations. Part 1. *Contr. Syst. and Comp.*, (1).

[19] M. I. Schlesinger and V. V. Giginyak. Solution to structural recognition (MAX,+)-problems by their equivalent transformations. Part 2. *Contr. Syst. and Comp.*, (2).

[20] E. Sudderth and M. Jordan. Shared Segmentation of Natural Scenes Using Dependent Pitman-Yor Processes. In *NIPS*, 08.

[21] Z. Tu and S.-C. Zhu. Image Segmentation by Data-Driven Markov Chain Monte Carlo. *PAMI*, 24(5), 2002.

[22] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. In *ICCV*, 2009.

[23] M. Wainwright, T. Jaakkola, and A. Willsky. MAP estimation via agreement on trees: Message-passing and linear-programming approaches. *IEEE Trans. Information Theory*, 51(11), 2005.

[24] T. Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In *CVPR*, 2008.

[25] O. J. Woodford, C. Rother, and V. Kolmogorov. A global perspective on map inference for low-level vision. In *Microsoft Research Technical Report*, 2009.

[26] A. Zalesny and L. V. Gool. A compact model for viewpoint dependent texture synthesis. In *SMILE Workshop, Lecture notes in computer science*, volume 2018, 2001.

[27] S.-C. Zhu, Y. Wu, and D. Mumford. FRAME: Filters, Random fields And Maximum Entropy— towards a unified theory for texture modeling. *IJCV*, 27:1–20, 1998.