Comparison of energy minimization algorithms for highly connected graphs

Vladimir Kolmogorov[1] and Carsten Rother[2]

[1] University College London; vnk@adastral.ucl.ac.uk
[2] Microsoft Research Ltd., Cambridge, UK; carrot@microsoft.com

Abstract. Algorithms for discrete energy minimization play a fundamental role for low-level vision. Known techniques include graph cuts, belief propagation (BP) and recently introduced tree-reweighted message passing (TRW). So far, the standard benchmark for their comparison has been a 4-connected grid-graph arising in pixel-labelling stereo. This minimization problem, however, has been largely solved: recent work shows that for many scenes TRW finds the global optimum. Furthermore, it is known that a 4-connected grid-graph is a poor stereo model since it does not take occlusions into account.

We propose the problem of stereo with occlusions as a new test bed for minimization algorithms. This is a more challenging graph since it has much larger connectivity, and it also serves as a better stereo model. An attractive feature of this problem is that increased connectivity does not result in increased complexity of message passing algorithms. Indeed, one contribution of this paper is to show that sophisticated implementations of BP and TRW have the same time and memory complexity as that of 4-connected grid-graph stereo.

The main conclusion of our experimental study is that for our problem graph cut outperforms both TRW and BP considerably. TRW achieves consistently a lower energy than BP. However, as connectivity increases the speed of convergence of TRW becomes slower. Unlike 4-connected grids, the difference between the energy of the best optimization method and the lower bound of TRW appears significant. This shows the hardness of the problem and motivates future research.

1 Introduction

Many early vision problems can be naturally formulated in terms of energy minimization where the energy function has the following form:

$$E(\mathbf{x}) = \sum_{p \in \mathcal{V}} D_p(x_p) + \sum_{(p,q) \in \mathcal{E}} V_{pq}(x_p, x_q) \ . \tag{1}$$

Set $\mathcal{V}$ usually corresponds to pixels; x_p denotes the label of pixel p which must belong to some finite set. For motion or stereo, the labels are disparities, while for image restoration they represent intensities. This energy is often derived in the context of Markov Random Fields [1]: unary terms D_p represent data likelihoods, and pairwise terms V_{pq} encode a prior over labellings. Energy minimization framework has been applied with great success to many vision applications such as stereo [2–7], image restoration [2], image segmentation [8], texture synthesis [9]. Algorithms for minimizing energy E are therefore of fundamental importance in vision. In this paper we consider three different

algorithms: Graph Cut, belief propagation (BP) and tree-reweighted message passing (TRW). For the problem of stereo matching these methods are among the best performing optimization techniques [10]. A comparison of their advantages and disadvantages is at the end of this section.

So far, comparison studies of these optimization methods have been rather limited in the sense that they only consider energy functions with a particular graph structure [11–14]. The algorithms have been tested on the energy function arising in stereo matching problem [2]. This energy is defined on a graph with a 4-neighborhood system, where nodes correspond to pixels in the left image. Occlusion are not modeled since this gives a more complex and highly connected graph structure. The comparison studies consistently concluded that the lowest energy is obtained by TRW, graph cuts come second and BP comes third [11–14]. Very recently, it has been shown [13] that TRW even achieves the global optimum for standard benchmark stereo pairs [10]. Consequently, this problem, which was considered to be very challenging a decade ago, has now largely been solved. The comparison studies also showed that the proposed energy gives large error statistics compared with state-of-the art methods, and consequently progress in this field can only be achieved by improving the energy formulation itself, as stated in [11, 13].

The main goal of this paper is to test how different optimization methods perform on graphs with larger connectivity. Our study has two motivations. First, such energy functions are becoming increasingly important in vision [3–7, 15]. They typically arise when we need to match two images while imposing regularization on the deformation field. Pixels (or features) in one image can potentially match to many pixels (features) in the other image, which yields a highly connected graph structure.

Our second motivation is to understand better intrinsic properties of different algorithms. One way to achieve this is to consider a very difficult problem: Algorithm's weaknesses then become more apparent, which may suggest ways of improving the method. It is known that the presence of short cycles in the graph makes the problem harder for message passing techniques. From this point of view, the problem that we are considering is much more challenging than 4-connected grid graphs. Another indicator of the difficulty of our problem will be shown by our experiments.

We choose the energy function arising in the problem of stereo with occlusions [4]. In this case there are nodes corresponding to pixels in the left and right image, and each node has $K + 4$ neighbors where K is the number of disparities. We propose this problem as a new challenging test bed for minimization algorithms. Our experiments also show that modeling occlusions gives a significantly better stereo model, since the energy of the ground truth is close to the energy of the best optimization method, and the value of the energy correlates with the error statistics derived from ground truth.

When applying BP or TRW to this energy, we immediately run into efficiency problems. There are K labels and $O(NK)$ edges, so a straightforward implementation would take $O(NK^2)$ memory and time for one iteration, even with the distance transform technique in [16]. By exploiting a special structure of the energy we show that both quantities can be reduced to $O(NK)$. Thus, we get the same complexity as that of message passing for the simple stereo problem without occlusions.

We have tested the three different optimization methods on six standard benchmark images [10]. The findings are different to the scenario of a 4-connected grid-graphs. For our problem graph cut clearly outperforms message passing techniques, i.e. TRW and BP, both in terms of lower energy and lower error rates wrt to ground truth.

It is worth mentioning that energy functions with similar graph structure were used in other methods for stereo with occlusions [6, 7]. In both approaches each pixel is connected to $O(K)$ pixels in the other image. The former uses graph cuts as a minimization algorithm, as the latter uses BP. However, [7] does not attempt to apply message passing to the original function. Instead, an iterative technique is used where in each iteration the energy function is approximated with a simpler one, and BP is then applying to a graph with 4-neighborhood system.

Let us compare the three optimization methods.

Graph cuts were introduced into computer vision in the 90's [17, 2] and showed a major improvement over previously used simulated annealing [1]. The strength of graph cuts is that for many applications it gives very accurate results, i.e. it finds a solution with very low energy. In fact, in some cases it even finds a *global* minimum [17, 18]. A major drawback of graph cuts, however, is that it can be applied only to a limited class of energy functions. There are different graph cut-based methods: Expansion move [2], swap move [2] or jump move [19]. Each has its own restrictions that come from the fact that binary minimization problems used in the "inner loop" must be *submodular*. Expansion move algorithm is perhaps the most powerful technique [14], but can be applied to a smaller set of functions than swap move. Following [4], we use expansion move version of the graph cut algorithm for the problem of stereo with occlusions.

The class of functions that graph cuts can handle covers many useful applications, but in some cases the energy falls outside this class, for example, in the super-resolution problem [20] This may also occur when parameters of the energy function are learned from training data [21]. In this case one can either approximate a non-submodular function with a submodular one [15], or use more general algorithms. Two of such algorithms are described below.

Belief propagation (BP). Max-product loopy belief propagation (BP) [22, 16] is a very popular technique for approximate inference. Unlike graph cuts, BP can be applied to any function of the form 1. Unfortunately, recent studies have shown that for a simple stereo problem it finds considerably higher energy than graph cuts [11, 23, 12, 13].

Tree-reweighted message passing (TRW) was recently introduced by Wainwright et al. [24]. Similar to BP it can be applied to any function of the form 1. However, there are several important differences. First, on a simple stereo problem it finds slightly lower energy than graph cuts [12]. Second, it maintains a lower bound on the energy that can be used to measure how close we are to the energy of an optimal solution. Third, there is a variant of TRW algorithm, called TRW-S, with certain convergence properties [12]. In contrast, no convergence guarantees are known for BP algorithm. For our comparison we use this variant of the TRW algorithm introduced in [12].

2 Background

We begin by introducing our notation. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with the set of vertices $\mathcal{V}$ and the set of edges $\mathcal{E}$. We assume that for each $p \in \mathcal{V}$ variable x_p takes values in some discrete set $\mathcal{L} = \{0, \dots, K-1\}$ where K is the number of labels[3].

Function $D_p(\cdot)$ in energy 1 is determined by K values. It is convenient to treat D_p as a vector of size $K \times 1$. Later we introduce other vectors of size $K \times 1$ (in particular, messages m). Notation $D' = D + m$ denotes the usual sum of two vectors, i.e. $D'(k) = D(k) + m(k)$ for any $k \in \mathcal{L}$.

2.1 Overview of message passing algorithms

We now give an overview of BP and TRW algorithms. They both maintain messages m_{pq} for directed edges $p \to q$, which are vectors of size $K \times 1$. The basic operation of the algorithms is *passing a message* from node p to its neighbor q. The effect of this operation is that message m_{pq} gets updated according to a certain rule (which is different for BP and for TRW).

An important choice that we have to make is the schedule of updating messages. There are many possible approaches; for example, [11] uses parallel (or synchronous) and *accelerated* schedules, and [12] uses *sequential* schedule. In this paper we use the latter one. One advantage of this schedule is that it requires half as much memory compared to other schedules. For TRW algorithm sequential schedule also has a theoretical advantage described in the end of this section.

Sequential schedule is specified by some ordering of nodes $i(p)$, $p \in \mathcal{V}$ (which can be chosen arbitrarily). During the forward pass, we process nodes in the order of increasing $i(p)$, and we send messages from node p to all its forward neighbors (i.e. nodes q with $i(q) > i(p)$). After that we perform similar procedure in the reverse direction (backward pass). A precise description of the algorithm is given in Fig. 1. Note that the operation of computing the minimum in step 1(b) can be computed efficiently, for many interaction potentials $V_{p,q}$, in time O(k) using distance transforms [16].

0. Set all messages to zero.
1. For nodes $p \in \mathcal{V}$ do the following operation in the order of increasing $i(p)$:
   (a) Aggregation: compute $\widehat{D}_p = D_p + \sum_{(q,p)\in\mathcal{E}} m_{qp}$
   (b) Propagation: for every edge $(p,q) \in \mathcal{E}$ with $i(p) < i(q)$ update message m_{pq} as follows:
   - Compute $D_{pq} = \gamma_{pq}\widehat{D}_p - m_{qp}$
   - Set $m_{pq}(x_q) := \min_{x_p}\{D_{pq}(x_p) + V_{pq}(x_p, x_q)\}$
2. Reverse the ordering: set $i(p) := |\mathcal{V}| + 1 - i(p)$.
3. Check whether a stopping criterion is satisfied; if yes, terminate, otherwise go to step 1.

Fig. 1. Sequential message passing algorithm. *Function $i : \mathcal{V} \to \{1, 2, \dots, |\mathcal{V}|\}$ gives the ordering of nodes. Weighting coefficient γ_{pq} is 1 for BP and a value in $(0, 1]$ for TRW (see text).*

Memory requirements. An important property of the sequential schedule is that for each edge (q, r) it is enough to store message in only one direction. Namely, suppose

[3] To simplify notation, we assumed that number of labels is the same of all nodes. Note that in general this is not required.

that $i(q) < i(r)$ and p is the node being processed. Then we store message m_{qr} if $i(q) < i(p)$, and message m_{rq} otherwise. The reverse messages are not needed since we update them before they are used. The same space in memory can be used for storing one of the two messages. The exact moment when m_{qp} gets replaced with m_{pq} is when edge (p, q) is processed in step 1(b).

The fact that memory requirements of message passing can be reduced by half was first noted in [16] for a special case (bipartite graphs and simulation of parallel schedule of updating messages). It was generalized to arbitrary graphs and larger class of schedules in [12].

Weighting coefficients. Both BP and TRW algorithms have the structure shown in Fig. 1. The difference between the two is that they use difference coefficients γ_{pq}. For BP algorithm we set $\gamma_{pq} = 1$. Next we describe how to choose these coefficients for TRW algorithm.

First we select set $\mathcal{T}$ of trees in graph $\mathcal{G}$ such that each edge is covered by at least one tree. We also select probability distribution over $\mathcal{T}$, i.e. function $\rho : \mathcal{T} \rightarrow (0, 1]$ such that $\sum_{T \in \mathcal{T}} \rho(T) = 1$. Set $\mathcal{T}$ and distribution ρ define coefficients γ_{pq} as follows: $\gamma_{pq} = \rho_{pq}/\rho_p$ where ρ_p and ρ_{pq} are the probabilities that tree T chosen under ρ contains node p and edge (p, q), respectively.

TRW and lower bound on the energy. As shown in [24], for any set of messages $\mathbf{m} = \{m_{pq} \mid (p \rightarrow q) \in \mathcal{E}\}$ it is possible to compute a lower bound on the energy, denoted as $\Phi_\rho(\mathbf{m})$. In other words, for any $\mathbf{m}$ and for any configuration $\mathbf{x}$ we have $\Phi_\rho(\mathbf{m}) \leq E(\mathbf{x})$. Function $\Phi_\rho(\mathbf{m})$ serves as a motivation for TRW: the goal of updating messages is to maximize $\Phi_\rho(\mathbf{m})$, i.e. to get the tightest bound on the energy.

In general, TRW algorithms in [24] do not always increase the bound - function $\Phi_\rho(\mathbf{m})$ may go down (and the algorithm may not converge). In contrast, sequential schedule proposed in [12] does have the property that the bound never decreases, assuming that the following condition holds: trees in $\mathcal{T}$ are *monotonic* chains, i.e. chains $T = (p_1, \ldots, p_m)$ such that sequence $(i(p_1), \ldots, i(p_m))$ is monotonic. The algorithm in Fig. 1 with this selection of trees is referred to as *sequential tree-reweighted message passing* (TRW-S).

Choosing solution. An important question is how to choose solution $\mathbf{x}$ given messages $\mathbf{m}$. The standard method is to choose label x_p for node p that minimizes $\widehat{D}_p(x_p)$ where $\widehat{D}_p = D_p + \sum m_{qp}$ and the sum is over edges $(q, p) \in \mathcal{E}$. However, it is often the case that $\widehat{D}_p(x_p)$ has several minima. In the case of TRW algorithm this is not surprising: if upon convergence all nodes had unique minimum, then it would give the *global* minimum of the energy, as shown in [24]. Clearly, we cannot expect this in general since otherwise we could solve arbitrary NP-hard problems.

To alleviate the problem of multiple minima, we use the same technique as in [12]. We assign variables to nodes p in the order given by $i(p)$. We select label x_p that minimizes $D_p(x_p) + \sum_{i(q)<i(p)} V_{qp}(x_q, x_p) + \sum_{i(q)>i(p)} m_{qp}(x_p)$.

2.2 Stereo with occlusions

In this section we review the energy function used in [4], adopting it to our notation. For simplicity we restrict our attention to the case of two rectified cameras.

The set of nodes contains pixels $\mathcal{V}^L$ in the left image and pixels $\mathcal{V}^R$ in the right image, so $\mathcal{V} = \mathcal{V}^L \cup \mathcal{V}^R$. Label x_p for pixel p denotes its disparity. We assume that

$x_p \in \mathcal{L} = \{0, \dots, K-1\}$ where K is the number of disparities. Pixel p with label k corresponds to some pixel q in the other image which we denote as $q = \mathcal{F}(p,k)$. Formally, coordinates of q are defined as follows:

$$(q_x, q_y) = \begin{cases} (p_x - k, p_y) & \text{if } p \in \mathcal{V}^L \\ (p_x + k, p_y) & \text{if } p \in \mathcal{V}^R \end{cases}$$

Note that $q = \mathcal{F}(p,k)$ implies $p = \mathcal{F}(q,k)$, and vice versa.

The energy function in [4] does not use unary data terms D_p; instead, all information is contained in pairwise terms V_{pq}. In order to describe them, first we need to define the set of edges $\mathcal{E}$. It contains edges of two types: *coherence* edges $\mathcal{E}^C$ and *stereo* edges $\mathcal{E}^S$ discussed below.

Coherence edges. These edges encode the constraint that disparity maps in the left and right images should be spatially coherent. Set $\mathcal{E}^C$ contains edges (p,q) where p, q are neighboring pixels in the same image defined, for example, using 4-neighborhood system.

For the purpose of comparison of minimization algorithms we used Potts terms V_{pq}:

$$V_{pq}(x_p, x_q) = \lambda_{pq} \cdot [x_p \neq x_q]$$

where $[\cdot]$ is 1 if its argument is true, and 0 otherwise. This term prefers piecewise constant disparity maps. To get better results, however, it might be advantageous to use terms that allow smooth variations, especially when the number of disparities is large. A good choice could be truncated linear term.

Stereo edges. Each pixel p (except for pixels at image boundary) has K incident edges connecting it to pixels $\mathcal{F}(p,0), \dots, \mathcal{F}(p,K-1)$ in the other image. To simplify notation, we denote edge (p,q) with $q = \mathcal{F}(p,k))$ as either (p,k) or (k,q).

Terms V_{pk} combine data and visibility terms defined in [4]. They can be written as

$$V_{pk}(x_p, x_q) = \begin{cases} M_{pk} & \text{if} \quad x_p = x_q = k \\ \infty & \text{if } x_p = k, x_q < k \\ & \text{or } x_q = k, x_p < k \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $q = \mathcal{F}(p,k)$ (see Fig. 2). Constant M_{pk} is the *matching score* between pixels p and q. The expansion move algorithm in [4] can only be applied if all scores are non-positive. Therefore, M_{pk} can be defined, for example, as $M_{pk} = \min\{||Intensity(p) - Intensity(q)||^2 - C, 0\}$ where C is a positive constant.

3 Efficient message passing for stereo with occlusions

In this paper we apply sequential message passing algorithm in Fig. 1 to the energy function defined in the previous section. However, a naive implementation is extremely inefficient. Indeed, consider first the memory requirements. We have $O(NK)$ edges where N is the number of pixels. For each edge we need to store a message which is a vector with K components. This results in $O(NK^2)$ memory requirements.

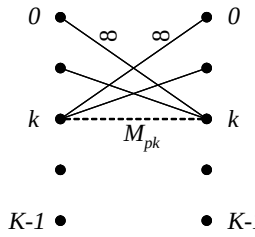


Fig. 2. Structure of term $V_{pk}(\cdot,\cdot)$ for stereo edge (p,q) with $q=\mathcal{F}(p,k)$. Left column represents pixel p, right column pixel q. Costs $V(k',k'')$ are shown on links from label k' to k''. The dashed link $k-k$ has cost M_{pk}, solid links have infinite costs. Links that are not shown have cost zero.

We now show how this number can be reduced to $O(NK)$. Consider message m_{pk} from pixel p to pixel $q = \mathcal{F}(p,k)$. It is obtained as the result of applying distance transform to vector D_{pk} via edge term V_{pk} (step 1(b) of the algorithm). Inspecting the structure of V_{pk} we conclude that

$$m_{pk}(k') = \begin{cases} A_{pk} = \min\{\widetilde{A}_{pk}, \widetilde{C}_{pk}\} & \text{if } k' < k \\ B_{pk} = \min\{\widetilde{B}_{pk} + M_{pk}, \widetilde{C}_{pk}\} & \text{if } k' = k \\ C_{pk} = \min\{\widetilde{A}_{pk}, \widetilde{B}_{pk}, \widetilde{C}_{pk}\} & \text{if } k' > k \end{cases}$$

where

$$\widetilde{A}_{pk} = \min_{0 \le k' < k} D_{pk}(k') \; ; \;\; B_{pk} = D_{pk}(k) \; ; \;\; \widetilde{C}_{pk} = \min_{k < k' < K} D_{pk}(k') \qquad (3)$$

Therefore, although m_{pk} is a vector with K components, it can be stored using only three numbers - A_{pk}, B_{pk} and C_{pk}. (In fact, even two numbers are sufficient. The messages are defined only up to an additive constant. Thus, it is enough to store $A_{pk} - C_{pk}$ and $B_{pk} - C_{pk}$, for example.)

To summarize, messages can be stored using $4NK$ numbers, ignoring effects at image boundaries ($2NK$ numbers for coherence edges and $2NK$ numbers for stereo edges[4]). We also need NK numbers to store matching scores M_{pk}.

Now let us consider the complexity of one iteration. If we are not careful, we may get $O(NK^2)$ running time even with the trick described above. Next we show how to implement the algorithm so that we get $O(NK)$ complexity for one iteration.

Aggregation. Let us consider the aggregation step for pixel p. We need to sum $K+4$ vectors of size K corresponding to K stereo edges and 4 coherence edges. A naive implementation would take $O(K^2)$ time. However, it is possible to reduce it to $O(K)$ using ideas from dynamic programming.

Summing messages in coherence edges is not a problem since their number is constant. Thus, we focus on summing messages corresponding to stereo edges, i.e. computing $D = \sum_{k=0}^{K-1} m_{kp}$. Suppose that message m_{kp} is described by numbers A_k, B_k,

[4] Recall that in the sequential message passing algorithm for each edge we need to store a message only in one direction.

C_k (we drop subscript p for brevity). We can write

$$D = \begin{pmatrix} B_0 \\ C_0 \\ \dots \\ C_0 \\ C_0 \end{pmatrix} + \begin{pmatrix} A_1 \\ B_1 \\ \dots \\ C_1 \\ C_1 \end{pmatrix} + \dots + \begin{pmatrix} A_{K-2} \\ A_{K-2} \\ \dots \\ B_{K-2} \\ C_{K-2} \end{pmatrix} + \begin{pmatrix} A_{K-1} \\ A_{K-1} \\ \dots \\ A_{K-1} \\ B_{K-1} \end{pmatrix}$$

To compute D, we first compute sums $\bar{A}_k = \sum_{k'=k+1}^{K-1} A_{k'}$ and $\bar{C}_k = \sum_{k'=0}^{k-1} C_{k'}$ for $k = 0, 1, \dots, K-1$ (by definition, $\bar{A}_{K-1} = \bar{C}_0 = 0$). This can be done in $O(K)$ time using recursions

$$\bar{A}_{k-1} = \bar{A}_k + A_k \ , \quad \bar{C}_{k+1} = \bar{C}_k + C_k \ .$$

Now computing D is easy: $D(k) = \bar{A}_k + B_k + \bar{C}_k$.

Propagation. Now consider the propagation step 2(b) for pixel p. Updating messages in coherence edges can be done in $O(K)$ time using distance transform techniques in [16]. We focus on updating messages m_{pk} in stereo edges for disparities $k \in \mathcal{L}$ with $i(\mathcal{F}(p,k)) > i(p)$. In order to update message m_{pk}, we need to compute numbers $\widetilde{A}_{pk} = \min_{0 \le k' < k} D_{pk}(k')$ and $\widetilde{C}_{pk} = \min_{k < k' < K} D_{pk}(k')$ (eq. 3). Direct calculation of these minima would take $O(K)$ time, resulting in $O(K^2)$ complexity for $O(K)$ stereo edges. To improve the running time, we do the following precalculation. For vector $\widehat{D}_p$ obtained after aggregation step 2(a), we compute values $\widehat{A}_{pk} = \min_{0 \le k' < k} \widehat{D}_p(k')$ and $\widehat{C}_{pk} = \min_{k < k' < K} \widehat{D}_p(k')$ (by definition, $\widehat{A}_{p0} = \widehat{C}_{p,K-1} = \infty$). This can be done in $O(K)$ time using recursions

$$\widehat{A}_{p,k+1} = \min\{\widehat{A}_{pk}, \widehat{D}_p(k)\}, \ \widehat{C}_{p,k-1} = \min\{\widehat{C}_{pk}, \widehat{D}_p(k)\}.$$

Now values $\widetilde{A}_{pk}$, $\widetilde{B}_{pk}$, $\widetilde{C}_{pk}$ can be computed in constant time as follows:

$$\widetilde{A}_{pk} = \gamma_{pk} \widehat{A}_{pk} - A_{kp} \ ; \ \ \widetilde{B}_{pk} = \gamma_{pk} \widehat{D}_p(k) - B_{kp} \ ; \ \ \widetilde{C}_{pk} = \gamma_{pk} \widehat{C}_{pk} - C_{kp} \qquad (4)$$

where A_{kp}, B_{kp}, C_{kp} describe message m_{kp} in the reverse direction.

4 Experimental results

We tested the methods on four benchmark stereo images, which were used in the stereo survey paper [10] and also two ground truth data sets with large disparity range [25]. All data sets are available online[5]. Fig. 3-6 show left disparity maps produced by different methods. Visually BP performed worse than graph cut and TRW, and also the results of BP were always less smooth. Numerical results for all six data sets are summarized in table 1. All experiments give a concise and clear message: Graph cut consistently outperforms TRW and BP, both in terms of lower energy and smaller error rate wrt ground truth ($B_{\bar{\mathcal{O}}}$ and $B_{\mathcal{D}}$). For smaller number of labels ($K < 30$) TRW clearly outperforms BP, otherwise TRW performs only marginally better. For all examples the quality of the results is correlated with the obtained energy, i.e. low energy corresponds to a low

[5] http://cat.middlebury.edu/stereo/

Image	Graph Cut			TRW			BP			Ground Truth
	$B_{\bar{\mathcal{O}}}$	$B_{\mathcal{D}}$	E	$B_{\bar{\mathcal{O}}}$	$B_{\mathcal{D}}$	E	$B_{\bar{\mathcal{O}}}$	$B_{\mathcal{D}}$	E	E (violation)
Tsukuba (K=16)	1.84	6.50	-1536	2.62	7.15	-1534	7.52	16.10	-1495	not available
Sawtooth (K=19)	0.56	6.26	-2071	0.65	7.12	-2065	3.43	10.39	-2020	-2027 (0.16%)
Venus (K=21)	1.20	6.11	-2118	1.55	8.12	-2109	10.31	14.88	-2021	-2069 (0.47%)
Map (K=29)	0.38	5.32	-3460	0.58	7.20	-3407	1.21	9.64	-3374	-3410 (0.40%)
Teddy (K=54)	13.14	23.35	-10273	14.88	26.95	-9889	15.25	27.63	-9834	not available
Cones (K=56)	5.16	11.99	-13936	6.04	14.16	-13648	9.25	15.14	-13455	not available

Table 1. Comparison table for six benchmark stereo pairs applied to the optimization methods: Graph cut, TRW and BP. *Both TRW and BP were run for* 10.000 *iterations, and graph cut until convergence. The values for* $B_{\bar{\mathcal{O}}}$ *and* $B_{\mathcal{D}}$ *correspond to the percentage of pixels in non-occluded (*$B_{\bar{\mathcal{O}}}$*) and textureless (*$B_{\mathcal{D}}$*) areas with a disparity error greater than* 1 *wrt ground truth. These are standard error measurements as proposed in [10]. Note that all energies* E *are scaled by* 10^{-3}*. The last column gives the energy of the ground truth. Note that a very small percentage of pixels in the ground truth image violate the visibility constrained, which are ignored for the computation of the ground truth energy. Furthermore, the energy of the ground truth can only be computed for three data sets since for Tsukuba only one ground truth disparity map is available and Teddy and Cones have undefined areas in the disparity map. (see text for discussion).*

error statistics ($B_{\bar{\mathcal{O}}}$ and $B_{\mathcal{D}}$). Also, the energy of the ground truth (last column table 1) lies within the range of the energy computed by graph cut and TRW. For stereo without occlusions these two observations could not be established: The energy of the ground truth is considerably larger than graph cut and BP, and low energy did not necessarily correspond to a good result [11, 13]. Therefore, we can conclude that modeling occlusions gives a better stereo model. The fact that the ground truth energy is larger than the best method does not contradict to this: The problem is inherently ambiguous, which means that it is impossible to design an energy function whose global minimum always gives a correct solution.

Plots of energy vs. runtime are shown in Fig. 7. For instance, one iteration of TRW takes about 3.26 sec. for teddy (image size 450×375 and $K = 54$) on a Pentium IV 3.2 GHz processor. For all examples the discrete curve for graph cut is always below the curve of TRW and BP. An interesting observation is that the relative performance of TRW and BP depends on the number of labels: Larger connectivity makes TRW algorithm much slower, while the speed of BP is affected less significantly. Note, however, that when TRW is run long enough, it always outperformed BP (see table 1). It is worth noting that neither TRW nor BP converged. BP gets into a loop after typically $50 - 200$ iterations. In case of TRW the lower bound never decreases with time. Since it is bounded from above, the lower bound must converge to a fixed number. In our experiments, however, the lower bound of TRW continued increasing slowly even after 50000 iterations (for Tsukuba), which means that the algorithm still did not converge.

In order to understand how difficult our problem is, we looked at how close the energy E_{min} of the best method is to the lower bound E_{bound} given by TRW. Since absolute numbers are not very meaningful, we can consider the ratio $\frac{E_{min} - E_{bound}}{E_{bound}}$. If all energy values are non-negative, then this ratio gives an upper bound on the approximation factor. In our case, however, the energy can be negative due to numbers $M_{pq} \leq 0$. To solve this problem, we added constant NC to the energy where N is the number of pixels and C is defined in section 2.2. Since there are at most N terms M_{pq}

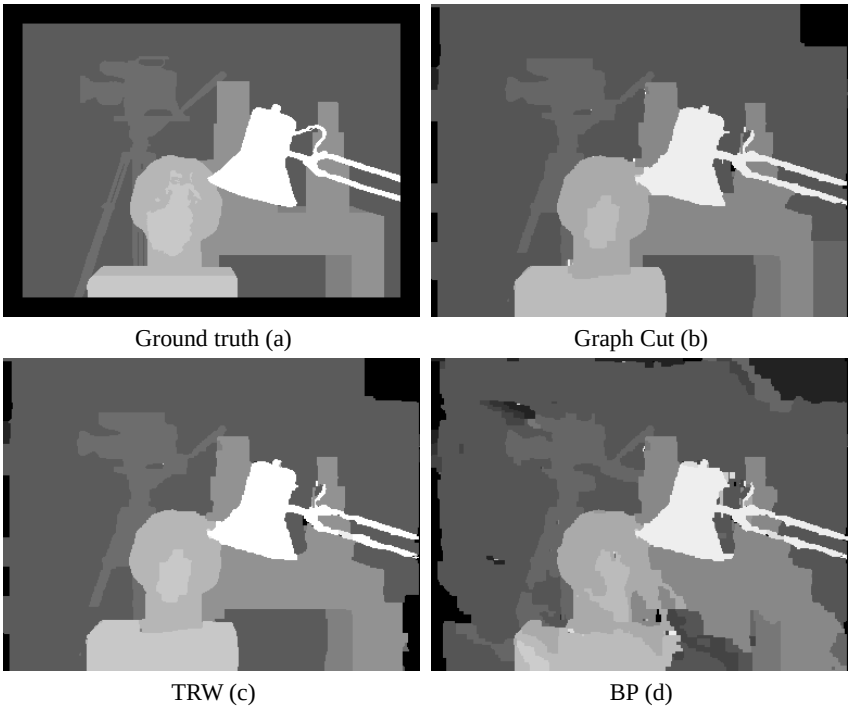


Fig. 3. Tsukuba image. *(a) Ground truth disparity for the left image (black pixels are unknown), (b) disparity map produced by graph cuts, (c) TRW, and (d) BP, which is clearly the worst result.*

in the energy and $M_{pq} \geq -C$, this ensures that energy is always non-negative. Furthermore, absolute energy values of the two models: stereo with and without occlusions are related, if we use same matching costs and similar smoothness parameters. This is confirmed by our experiments: E_{min} differ by about 3 times for the two models and Tsukuba data set, i.e. they are of the same order of magnitude. For stereo without occlusions ratios $\frac{E_{min}-E_{bound}}{E_{bound}}$ were as follows [12]: Tsukuba (0.0037%), Map (0.055%), Sawtooth (0.096%), and Venus(0.014%). For our model the corresponding values are: Tsukuba (3.09%), Map (3.28%), Sawtooth (1.27%), and Venus(2.26%). These values are in average two to three orders of magnitude larger for our model. Consequently, we may conclude that our problem is considerably harder than stereo without occlusions.

4.1 Settings for TRW

In order to implement TRW-S algorithm we need to make several choices. First, we need to select the ordering of nodes $i(p)$. In our implementation we used row-major order for both left and right images, and nodes of the left images had smaller ordering than nodes of the right image. Next, we need to choose the set of trees $\mathcal{T}$. As described in sec. 2, these trees must be chains that are monotonic with respect to ordering $i(p)$. We selected each horizonal and vertical line in the two images as a single chain; we call them *coherence chains*[6]. In addition, every stereo edge was declared to be a chain.

[6] There are $2(W + H)$ such chains where W is the width of the image and H is the height.

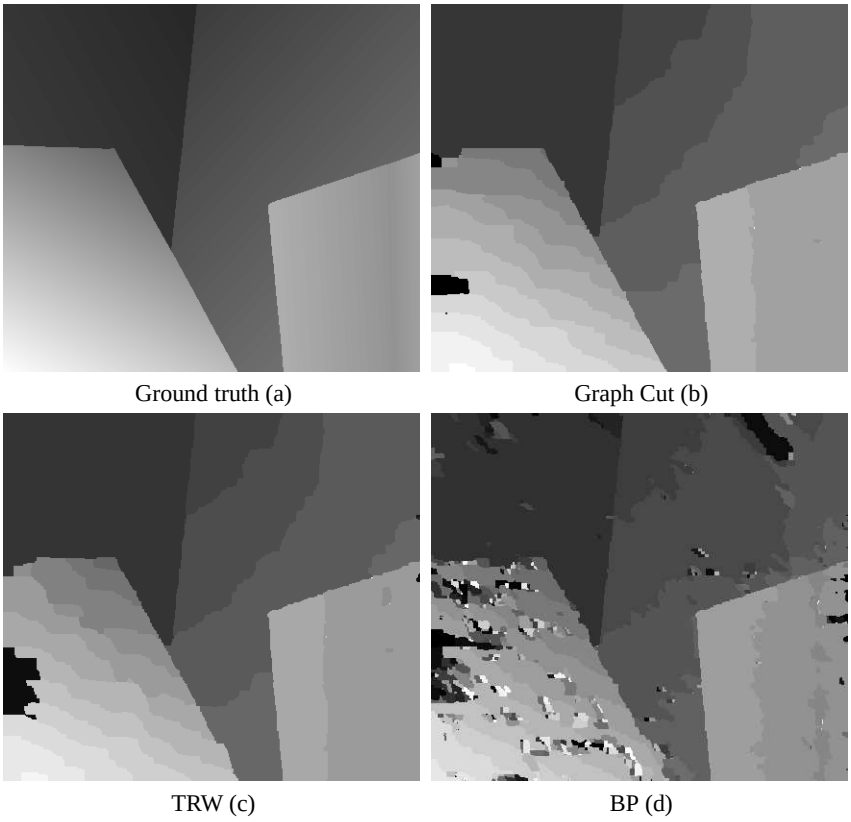


Fig. 4. Venus image. *(a) Ground truth disparity for the left image, (b) disparity map produced by graph cuts, which has the lowest error statistics, (c) TRW, and (d) BP.*

It can be seen that with this choice every edge in the graph is covered by exactly one tree. Finally, we need to select probability distribution ρ^T over trees $T \in \mathcal{T}$. As our experiments show, this distribution affects the results of the algorithm significantly.

Intuitively, coherence and stereo chains are quite different, therefore they should be assigned different probabilities. The difference between coherence and stereo chains, however, is not the only source of asymmetry. Indeed, consider some node p and an incident stereo edge (p, q) where $q = \mathcal{F}(p, k)$. Term V_{pk} for this edge has a very special structure; in particular, there is one preferred label, namely label k. Recall that if labels of pixels p and q are k then this edge contributes matching cost M_{pq} to the energy function, otherwise the penalty is either 0 or ∞. Thus, it could be beneficial to select probabilities that would favor label k over other labels $k' \in \mathcal{L} - \{k\}$ for the chain corresponding to edge (p, q), and we will show that this improves the performance of TRW. Since the scheme described in sec. 2 does not allow this (each tree has a single probability which does not depend on labels), we now extend the tree-reweighted algorithm to allow probabilities that depend on labels. Consider the case when each edge is covered by exactly one chain. Let us define a probability distribution over trees

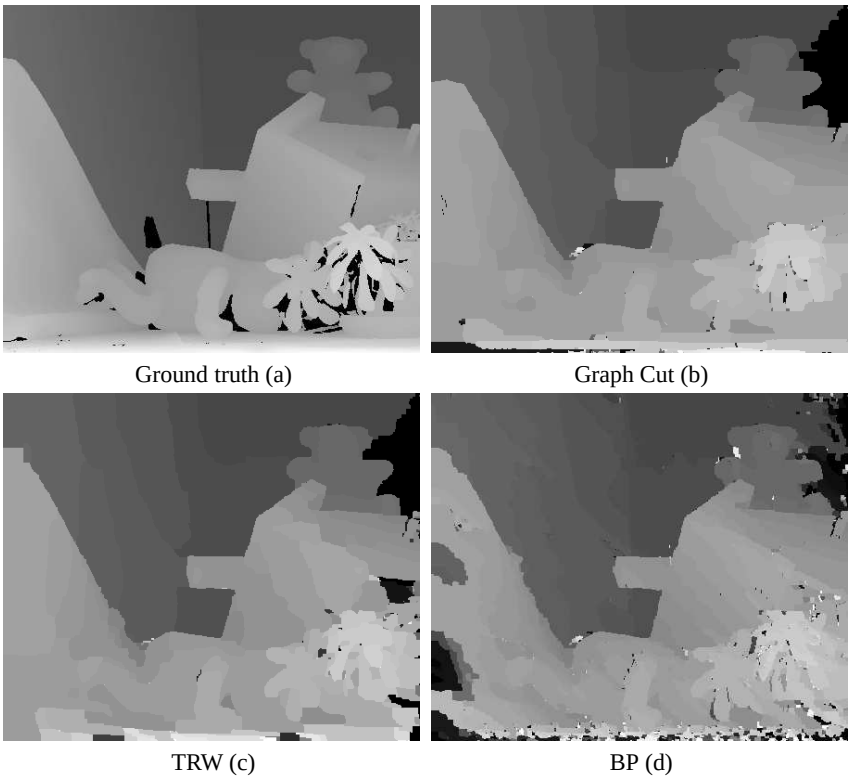


Fig. 5. Teddy image. *(a) Ground truth disparity for the left image (black pixels are unknown), (b) disparity map produced by graph cuts, which has the lowest error statistics, (c) TRW, and (d) BP.*

for each node $p \in \mathcal{V}$ *and label* $k \in \mathcal{L}$. We denote it as $\rho(T; p, k)$. We require that $\sum_{T \in \mathcal{T}} \rho(T; p, k) = 1$ for all p, k. In addition, $\rho(T; p, k)$ must be positive if tree T contains node p, and zero otherwise. Using these probabilities, we define coefficients $\gamma_{pq}(k)$ as follows: $\gamma_{pq}(k) = \rho(T; p, k)$ where T is the tree containing edge (p, q). The algorithm in Fig. 1 is then modified as follows: In step 1(b) vector D_{pq} is computed as $D_{pq}(k) = \gamma_{pq}(k)\widehat{D}_p(k) - m_{qp}(k)$ for all $k \in \mathcal{L}$. We claim that the modified algorithm has the same properties properties as the sequential tree-reweighted message passing method in [12]. In particular, the lower bound is guaranteed not to decrease, and there exists a limit point satisfying the weak tree agreement condition (see appendix A).

Let us apply this scheme to the problem of stereo with occlusions. Consider node $p \in \mathcal{V}$ and label $k \in \mathcal{L}$. This node is contained in $K + 2$ trees (unless it is a pixel near the image boundary): vertical coherence chain, horizontal coherence chain and K

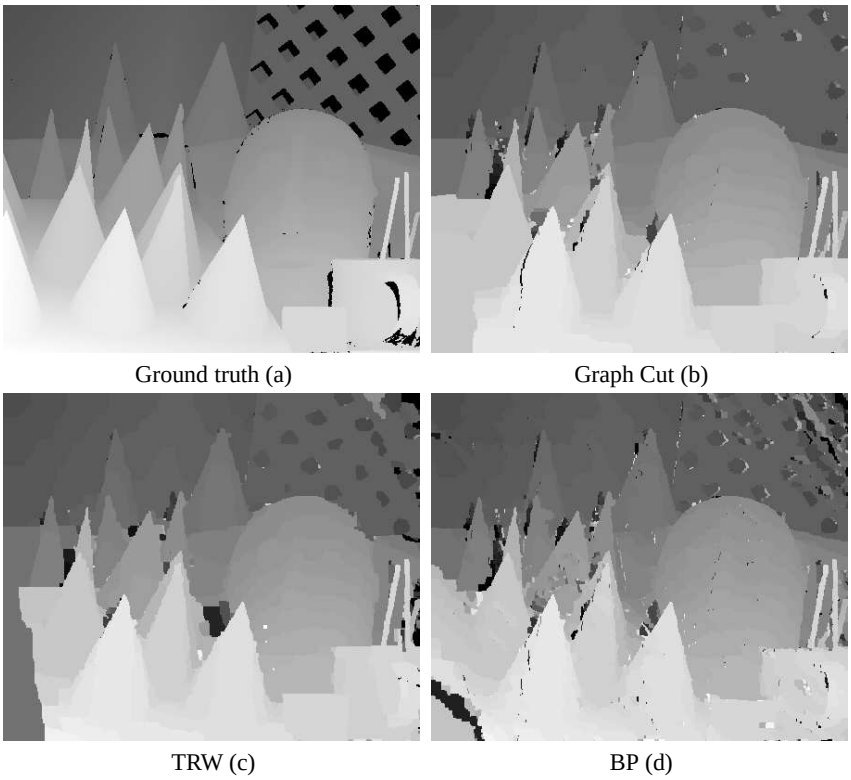


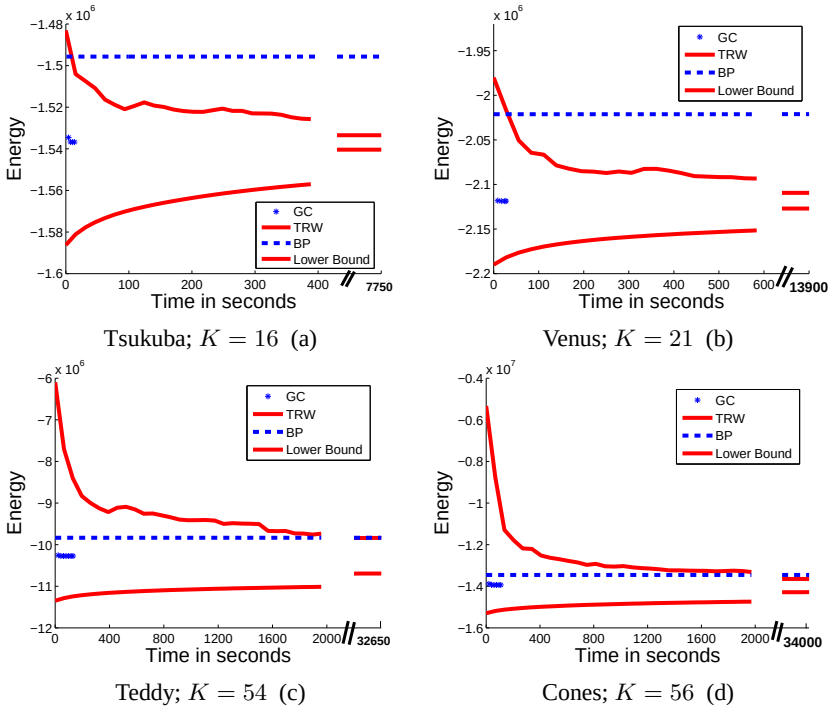
Fig. 6. Cones image. *(a) Ground truth disparity for the left image (black pixels are unknown), (b) disparity map produced by graph cuts, which has the lowest error statistics, (c) TRW, and (d) BP.*

stereo chains. We set probabilities $\rho(T;p,k)$ as follows:

$$\rho(T;p,k) = \begin{cases} \rho^C & \text{if } T \text{ is a coherence edge} \\ \rho^{S1} & \text{if } T = (p, \mathcal{F}(p,k)) \\ \rho^{S2} & \text{if } T = (p, \mathcal{F}(p,k')) \text{ for } k' \neq k \end{cases} .$$

Note that there must hold $2\rho^C + \rho^{S1} + (K-1)\rho^{S2} = 1$. Due to this constraint we are left with two degrees of freedom for the choice of the tree probabilities: ρ^C and $\beta^S = \rho^{S1}/\rho^{S2}$. Note that in the TRW algorithm the γ_{pk} in eqn. 4 has to be replaced by: $\gamma_{pk} = \rho^{S1}$ for $\widetilde{B}_{pk}$ and $\gamma_{pk} = \rho^{S2}$ for $\widetilde{A}_{pk}$ and $\widetilde{C}_{pk}$.

We examined different settings of ρ^C and β^S for three data sets. We discovered that the settings depend on the number of labels. For a thorough investigation we re-scaled the teddy image with a factor of 1.5 and 3 ("Teddy Small"), which correspond to a maximum disparity of 36 and 18 respectively. Fig. 8 shows the energy of TRW for a large range of values for ρ^C and β^S, where TRW was run for a fixed amount of 700 iterations. An obvious observation is that for extreme settings, e.g. β^S very close to 1 or below 0.4, the results are worse. The first conclusion we can draw is that the energy is



Tsukuba; $K = 16$ (a) Venus; $K = 21$ (b)

Teddy; $K = 54$ (c) Cones; $K = 56$ (d)

Fig. 7. Comparison of energies and lower bound with respect to runtime. *(discussion in text).*

more sensitive to parameters settings for larger disparities. For "Teddy Small" the range of comparable low energies for ρ^C is $[0.4, 0.9]$ whereas for teddy it is $[0.7, 0.8]$. The second observation is that parameters which give the lowest energy differ, depending on the number of disparities. The optimal setting of ρ^C is 0.76 $(K = 54)$ and 0.9 $(K = 36$ and $18)$. The optimal probability for different stereo edges β^S is less sensitive to the number of disparities. For these three examples a value of $\beta^S = 3.0$ gives low energy. Taking this into account we chose the settings as follows: $\rho^C = 0.9 (K < 40)$; otherwise 0.78; and $\beta^S = 3.0$. We do not claim that this is the optimal setting for TRW for this type of energy, however, we believe that it is sufficient for a comparison to other methods. We believe that further testing of these probabilities might improve the performance of TRW only marginal. A more significant improvement might come from changing the structure of the trees, e.g. choosing longer stereo chains.

5 Conclusions

We have presented an experimental comparison of three optimization techniques: Graph cut, BP and TRW for highly connected graphs. We have chosen the energy of the stereo with occlusions problem. Despite high connectivity of the graph, we have shown that message passing techniques can still be applied efficiently.

In the past comparisons have only been carried out for relatively simple 4-connected grid-graphs, in particular for stereo without occlusions. Our findings are different to 4-connected graphs where TRW outperforms graph cut, and even achieves the global

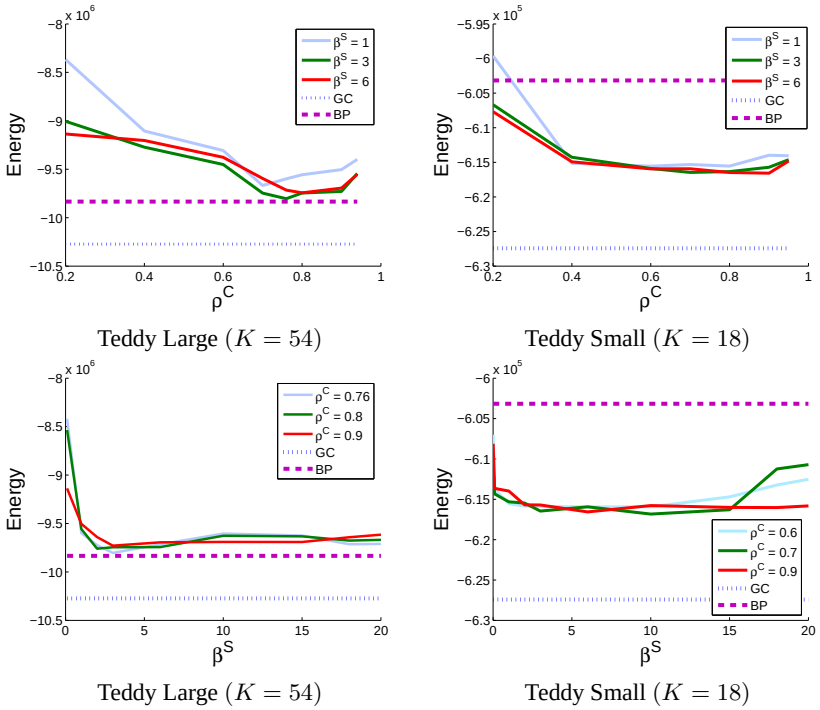


Fig. 8. Testing TRW settings *for the teddy data set with* 54 *(left column) and* 18 *(right column) disparities. (discussion in text).*

optimum for some problems. For highly connected graphs, graph cut clearly outperforms TRW and BP, both in terms of lower energy and lower error rates with respect to ground truth. We found that for all examples TRW is capable of obtaining lower energy than BP. However, as the connectivity increases, the speed of convergence for TRW becomes slower and slower, while the speed of BP is affected less significantly. This suggests that a future direction of research is to try improving the speed of TRW, like by choosing trees in a different way or using a different schedule of updating messages. We believe that if the speed is improved then TRW may still outperform graph cuts.

The experiments show that modeling occlusions gives a better stereo model. Another finding is that the difference between the lower bound of TRW and the minimum energy of the best method is significant compared to 4-connected graphs. This indicates the hardness of the problem, at least for algorithms based on solving LP relaxation (such as TRW). Consequently we propose this energy as new test bed for optimization techniques and hope that it will motivate future research in this area. Furthermore, we also plan to analyse other vision problems with highly connected graphs such as [15].

A Guarantee on lower bound

Let $\bar{\theta}$ be the parameter vector of the original energy function, i.e. $\bar{\theta}_p(k) = D_p(k)$, $\bar{\theta}_{pq}(k,k') = V_{pq}(k,k')$. Messages $\mathbf{m}$ define parameter vector θ^T for tree $T \in \mathcal{T}$ as

follows:

$$\theta_p^T(k) = \rho(T;p,k)(\bar{\theta}_p(k) + \sum\nolimits_{(p,q)\in\mathcal{E}} m_{qp}(k))$$
$$\theta_{pq}^T(k,k') = \bar{\theta}_{pq}(k,k') - m_{pq}(k') - m_{qp}(k)$$

The message passing algorithm described above is then equivalent to the sequential algorithm in [12] where the "node averaging" operation is performed as follows:

1. Compute $\tilde{\theta}_p = \sum_{T\in\mathcal{T}_p} \theta_p^T$.
2. Set $\theta_p^T(k) = \rho(T;p,k)\tilde{\theta}_p(k)$.

The proof that the bound $\sum_T \Phi(\theta^T)$ never decreases and there exists a limit point satisfying WTA condition now proceeds in exactly the same way as in [12].

References

1. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Machine Intell. **6** (1984) 721–741
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence **23(11)** (2001)
3. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: IEEE International Conference on Computer Vision. (2001)
4. Kolmogorov, V., Zabih, R.: Multi-camera scene reconstruction via graph cuts. In: Proc. Europ. Conf. Comp. Vision. Volume 3. (2002) 82–96
5. Sun, J., Zheng, N., Shum, H.: Stereo matching using belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence **25(7)** (2003) 787–800
6. Lin, M., Tomasi, C.: Surfaces with occlusions from layered stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **26(8)** (2004) 710–717
7. Sun, J., Li, Y., Kang, S.B., Shum, H.: Symmetric stereo matching for occlusion handling. In: IEEE Conf. on Comp. Vis. and Pat. Recog. (2005)
8. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: Proc. Int. Conf. Comp. Vision. (2001)
9. Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: Image and video synthesis using graph cuts. ACM Transactions on Graphics, SIGGRAPH 2003 (2003)
10. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Computer Vision **47** (2002) 7–42
11. Tappen, M.F., Freeman, W.T.: Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In: Proc. Int. Conf. Comp. Vision. (2003)
12. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. Technical Report MSR-TR-2005-38 (2005) Earlier version appeared in AISTATS 2005.
13. Meltzer, T., Yanover, C., Weiss, Y.: Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In: Proc. Int. Conf. Comp. Vision. (2005)
14. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for markov random fields. In: Proc. Europ. Conf. Comp. Vision. (2006)
15. Rother, C., Kumar, S., Kolmogorov, V., Blake, A.: Digital tapestry. In: IEEE Conf. on Comp. Vis. and Pat. Recog. (2005)

16. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. In: IEEE Conf. on Comp. Vis. and Pat. Recog. (2004)
17. Greig, D., Porteous, B., Seheult, A.: Exact maximum a posteriori estimation for binary images. Journal of the Royal Statistical Society, Series B **51** (1989) 271–279
18. Ishikawa, H.: Exact optimization for Markov Random Fields with convex priors. IEEE Trans. Pattern Anal. Machine Intell. **25(10)** (2003) 1333–1336
19. Veksler, O.: Efficient graph-based energy minimization methods in computer vision. PhD thesis, Cornell University, Dept. of Computer Science, Ithaca, NY (1999)
20. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. Int. J. Computer Vision **40** (2000) 25–47
21. Kumar, S., Herbert, M.: Discriminative fields for modeling spatial dependencies in natural images. In: Advances in Neural Information Processing Systems. (2004)
22. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc. (1988)
23. Barbu, A., Yuille, A.L.: Motion estimation by Swendsen-Wang cuts. In: CVPR. (2004)
24. Wainwright, M., Jaakkola, T., Willsky, A.: MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. IEEE Transactions on Information Theory **51(11)** (2005) 3697–3717
25. Scharstein, D., Szelsiki, R.: High-accuracy stereo depth maps using structured light. In: IEEE Conf. on Comp. Vis. and Pat. Recog. (2003)