A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors

Richard Szeliski, *Fellow*, *IEEE*, Ramin Zabih, *Senior Member*, *IEEE*, Daniel Scharstein, *Member*, *IEEE*, Olga Veksler, *Member*, *IEEE*, Vladimir Kolmogorov, *Member*, *IEEE*, Aseem Agarwala, *Member*, *IEEE*, Marshall Tappen, *Member*, *IEEE*, and Carsten Rother, *Member*, *IEEE*

Abstract—Among the most exciting advances in early vision has been the development of efficient energy minimization algorithms for pixel-labeling tasks such as depth or texture computation. It has been known for decades that such problems can be elegantly expressed as Markov random fields, yet the resulting energy minimization problems have been widely viewed as intractable. Recently, algorithms such as graph cuts and loopy belief propagation (LBP) have proven to be very powerful: For example, such methods form the basis for almost all the top-performing stereo methods. However, the trade-offs among different energy minimization algorithms are still not well understood. In this paper, we describe a set of energy minimization benchmarks and use them to compare the solution quality and runtime of several common energy minimization algorithms. We investigate three promising recent methods—graph cuts, LBP, and tree-reweighted message passing—in addition to the well-known older iterated conditional mode (ICM) algorithm. Our benchmark problems are drawn from published energy functions used for stereo, image stitching, interactive segmentation, and denoising. We also provide a general-purpose software interface that allows vision researchers to easily switch between optimization methods. The benchmarks, code, images, and results are available at http://vision.middlebury.edu/MRF/.

Index Terms—Performance evaluation, Markov random fields, global optimization, graph cuts, belief propagation.

1 Introduction

Many problems in early vision involve assigning each pixel a label, where the labels represent some local quantity such as disparity. Such pixel-labeling problems are naturally represented in terms of energy minimization, where the energy function has two terms: one term penalizes solutions that are inconsistent with the observed data, whereas the other term enforces spatial coherence (piecewise smoothness). One of the reasons why this framework is so popular is that it can be justified in terms of maximum a posteriori estimation of a Markov random field (MRF) [5], [19]. Despite the elegance and power of the energy minimization approach, its early adoption was slowed by computational considerations. The algorithms that were originally used, such as iterated conditional modes (ICM) [5] or simulated annealing [4], proved to be either ineffective or extremely inefficient.

Over the last few years, energy minimization approaches have had a renaissance, primarily due to powerful new optimization algorithms such as graph cuts [11], [31] and loopy belief propagation (LBP) [50]. The results, especially in stereo, have been dramatic; according to the widely used Middlebury stereo benchmarks [42], almost all the top-performing stereo methods rely on graph cuts or LBP. Moreover, these methods give substantially more accurate results than were previously possible. Simultaneously, the range of applications of pixel-labeling problems has also expanded dramatically, moving from early applications such as image restoration [5], texture modeling [20], image labeling [13], and stereo matching [4], [11] to applications such as interactive photo segmentation [8], [39] and the automatic placement of seams in digital photomontages [1].

Relatively little attention has been paid, however, to the relative performance of various optimization algorithms. Among the few exceptions are [9], which compared the efficiency of several different max-flow algorithms for graph cuts, and [45], which compared graph cuts with LBP for stereo. Tappen and Freeman [45] also noted a particularly impressive demonstration of the effectiveness of modern energy minimization methods: For the stereo problems in the Middlebury benchmarks, both graph cuts and LBP produced results

- *R. Szeliski is with Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399. E-mail: szeliski@microsoft.com.*
- *R. Zabih is with the Department of Computer Science, Cornell University, 4130 Upson Hall, Ithaca, NY 14853. E-mail: rdz@cs.cornell.edu.*
- *D. Scharstein is with the Department of Computer Science, Middlebury College, Middlebury, VT 05753. E-mail: schar@middlebury.edu.*
- *O. Veksler is with the Computer Science Department, Middlesex College 361, University of Western Ontario London ON N6A 5B7 Canada. E-mail: olga@csd.uwo.ca.*
- *V. Kolmogorov is with the Adastral Park Campus, University College London, Adastral Park, Martlesham Heath, IP5 3RE, UK. E-mail: vnk@adastral.ucl.ac.uk.*
- *A. Agarwala is with the Adobe Systems, Inc., 801 N 34th St., Seattle, WA 98103. E-mail: aseem@agarwala.org.*
- *M. Tappen is with the Electrical and Computer Engineering, ENGR 3-230, University of Central Florida, Orlando, FL 32816-2450. E-mail: mtappen at cs.ucf.edu.*
- *C. Rother is with Microsoft Research Ltd, 7 JJ Thomson Avenue, Cambridge CB30FB, UK. E-mail: carrot@microsoft.com.*

Manuscript received 8 May 2007; revised 15 Nov. 2007; accepted 19 Nov. 2007; published online 20 Dec. 2007.
Recommended for acceptance by P. Torr.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-05-0266.
Digital Object Identifier no. 10.1109/TPAMI.2007.70844.

0162-8828/08/$25.00 © 2008 IEEE Published by the IEEE Computer Society

whose energy is lower than the energy of the ground-truth solution, indicating the need for better models and energy functions. We will return to this issue at the end of this paper.

Although it is generally accepted that algorithms such as graph cuts are a huge improvement over older techniques such as simulated annealing, less is known about the efficiency versus accuracy trade-off among more recently developed algorithms. In this paper, we propose a number of benchmark problems for energy minimization and use these benchmarks to evaluate and compare a number of different energy minimization algorithms for pixel-labeling problems. Since much of the work in energy minimization has been motivated by pixel-labeling problems over 2D grids, we have restricted our attention to problems with this simple topology. The extension to more general topologies such as 3D is straightforward but is not part of this work.

Concurrently with our work, Kolmogorov and Rother [28] compared tree-reweighted message passing (TRW), LBP, and graph cuts for a highly connected graph corresponding to the stereo model with occlusions [30] and found that graph cuts clearly outperform the other two methods. This differs from our results in this paper for 4-connected grids, which have much smaller connectivity.

This paper is organized as follows: In Section 2, we give a precise description of the energy functions that we consider and present a simple but general software interface to describe such energy functions and to call an arbitrary energy minimization algorithm. In Section 3, we describe the different energy minimization algorithms that we have implemented, and in Section 4, we present our set of benchmarks. In Section 5, we provide our experimental comparison of the different energy minimization methods, and in Section 6, we discuss our results. We conclude in Section 7.

2 Problem Formulation and Experimental Infrastructure

2.1 Energy Model

We define a pixel-labeling problem as assigning to every pixel p a label, which we write as l_p. The collection of all pixel-label assignments is denoted by l, the number of pixels is n, and the number of labels is m. The energy function E, which can also be viewed as the log likelihood of the posterior distribution of an MRF [19], [35], is composed of a data energy E_d and a smoothness energy E_s

$$E = E_d + \lambda E_s. \tag{1}$$

The data energy is simply the sum of a set of per-pixel data costs $d_p(l)$

$$E_d = \sum_p d_p(l_p). \tag{2}$$

In the MRF framework, the data energy comes from the (negative) log likelihood of the measurement noise.

We assume that pixels form a 2D grid, so that each p can also be written in terms of its coordinates $p = (i, j)$. We use the standard 4-connected neighborhood system, so that the smoothness energy is the sum of spatially varying horizontal and vertical nearest neighbor smoothness costs, $V_{pq}(l_p, l_q)$, where if $p = (i, j)$ and $q = (s, t)$, then $|i - s| + |j - t| = 1$. If we let $\mathcal{N}$ denote the set of all such neighboring pixel pairs, the smoothness energy is

$$E_s = \sum_{\{p,q\} \in \mathcal{N}} V_{pq}(l_p, l_q). \tag{3}$$

Note that in (3), the notation $\{p, q\}$ stands for an unordered set, that is, the sum is over *unordered* pairs of neighboring pixels.

In the MRF framework, the smoothness energy comes from the negative log likelihood of the prior. In this paper, we consider a general form of the smoothness costs, where different pairings of adjacent labels can lead to different costs. This is important in a number of applications, for example, image stitching and texture quilting [1], [15], [34].

A more restricted form of the smoothness energy is

$$E_s = \sum_{\{p,q\} \in \mathcal{N}} w_{pq} \cdot V(|l_p - l_q|), \tag{4}$$

where the smoothness terms are the product of spatially varying per-pairing weights w_{pq} and a nondecreasing function of the label difference $V(\Delta l) = V(|l_p - l_q|)$. Such energy functions typically arise in stereo matching [11] and image denoising. Although we could represent V using an m-valued lookup table, for simplicity, we instead parameterize V using a simple clipped monomial form

$$V(\Delta l) = \min(|\Delta l|^k, V_{\max}) \tag{5}$$

with $k \in \{1, 2\}$. If we set $V_{\max} = 1.0$, we get the Potts model, $V(\Delta l) = 1 - \delta(\Delta l)$, which penalizes any pair of different labels uniformly (δ is the unit impulse function).

Although they are not our primary focus, a number of important special cases have fast exact algorithms. If there are only two labels, the natural Potts model smoothness cost (which is called Ising model in this case) can be solved exactly with graph cuts [22]. This was first applied to images in [21]. If the labels have a linear ordering (for example, consecutive integers) and the smoothness cost is an arbitrary convex function, the work in [24] gives a graph-cut construction for an exact solution. An algorithm due to [23] yields an exact solution for linear smoothness $V(\Delta l) = \Delta l$ and convex data costs. However, the NP-hardness result proved in [11] applies if there are more than two labels, as long as the class of smoothness costs includes the Potts model. This, unfortunately, implies that the vast majority of MRF-based energy functions are intractable.

The class of energy functions we consider is quite broad, and not all energy minimization methods can handle the entire class. For example, acceleration techniques based on distance transforms [16] can significantly speed up message-passing algorithms such as LBP or TRW, yet these methods are only applicable for certain smoothness costs V. Other algorithms such as graph cuts only have good theoretical guarantees for certain choices of V (see Section 3 for a discussion of this issue). In this paper, we assume that any algorithm can run on any benchmark problem; this can generally be ensured by reverting to a weaker or slower version of the algorithm if necessary for a particular benchmark.

2.2 Software Interface for Energy Minimization

Now that we have defined the class of energy functions that we minimize, we need to compare different energy

```
// Abstract definition of an energy function E
EnergyFunction *E = new EnergyFunction(data, smooth);

// Energy minimization of E via ICM
solver = (MRF *) new ICM(nX, nY, nLabels, E);
// To minimize E via graph cuts instead, use this:
// solver = (MRF *) new Expansion(nX, nY, nLabels, E);

// Run one iteration, store the time taken in t
solver->optimize(1, &t);

// Print out the resulting energy and runtime
print_stats(solver->totalEnergy(), t);
```

Fig. 1. Sample code calling our energy minimization API.

minimization methods on the same energy function E. Conceptually, it is easy to switch from one energy minimization method to another, but in practice, most applications are tied to a particular choice of E. As a result, almost no one in vision has ever answered questions like "how would your results look if you used LBP instead of graph cuts to minimize your E?" (The closest to this was [45], which compared LBP and graph cuts for stereo.) In order to create a set of benchmarks, it was necessary to design a standard software interface (API) that allows a user to specify an energy function E and to easily call a variety of energy minimization methods to minimize E.

The software API is available on the project Web page at http://vision.middlebury.edu/MRF/, as are implementations of all of the energy minimization methods discussed in this paper and all of our benchmarks. The API allows the user to define any energy function described above. The data cost energy can be specified implicitly as a function $d_p()$ or explicitly as an array. The smoothness cost likewise can be specified either by defining the parameters k and $V_{\max}$ or by providing an explicit function or array. Excerpts from an example program that uses our API to call two different energy minimization algorithms on the same energy function are shown in Fig. 1.

Note that the interface also has the notion of an iteration, but it is up to each energy minimization method to interpret this notion. Most algorithms have some natural intermediate point where they have a current answer. By supporting this, our API allows us to plot the curve of energy versus time. This is particularly important because a number of powerful methods (such as TRW and graph cuts) make very small changes in the last few iterations.

It is, of course, more efficient to "hard-wire" a vision algorithm to make use of a particular energy minimization technique than to use our general-purpose API. However, the gains in efficiency appear to be quite modest, assuming that the same representation is used for the energy function. The major difference is that the API relies on virtual functions, which result in a slowdown of approximately 10 percent in the speed of function calls [14]. In practice, only a modest amount of time is spent in function calls, so the overhead of using the API is much smaller than this.

2.3 Evaluation Methodology

To evaluate the quality of a given solution, we need the final energy E along with the computation time required, as a function of the number of iterations. For every benchmark, we produce a plot that keeps track of the energy versus computation time for every algorithm tested.

We implemented the algorithms in C or C++ and ran the benchmarks on a modern Pentium 4. All of the experiments were run on the same machine (3.4 GHz and 2 Gbyte RAM).

Note that although the natural way to compare energy minimization algorithms is in terms of their energy and speed, it is not always the case that the lowest energy solution is the best one for a vision problem. We return to this issue in Section 6.

3 Energy Minimization Algorithms

In this section, we describe the optimization algorithms that we have implemented and included in our interface. Most of the energy minimization algorithms were implemented by their original inventors; the exceptions are ICM and LBP (for LBP, we received help from several experts).

3.1 Iterated Conditional Modes (ICM)

Iterated conditional modes [5] uses a deterministic "greedy" strategy to find a local minimum. It starts with an estimate of the labeling, and then, for each pixel, it chooses the label giving the largest decrease of the energy function. This process is repeated until convergence, which is guaranteed to occur, and, in practice, is very rapid.

Unfortunately, the results are extremely sensitive to the initial estimate, especially in high-dimensional spaces with nonconvex energies (such as those that arise in vision) due to the huge number of local minima. In our experiments, we initialize ICM in a winner-take-all manner, by assigning each pixel the label with the lowest data cost. This results in significantly better performance.

3.2 Graph Cuts

The two most popular graph-cut algorithms, called the *swap-move* algorithm and the *expansion-move* algorithm, were introduced in [11]. Both algorithms work by repeatedly computing the global minimum of a binary labeling problem in their inner loops. This process converges rapidly and results in a strong local minimum, in the sense that no "permitted move" will produce a labeling with lower energy.

For a pair of labels α, β, a swap move takes some subset of the pixels currently given the label α and assigns them the label β and vice versa. The swap-move algorithm finds a local minimum such that there is no swap move, for any pair of labels α, β, that will produce a lower energy labeling. Analogously, we define an expansion move for a label α to increase the set of pixels that are given this label. The expansion-move algorithm finds a local minimum such that no expansion move, for any label α, yields a labeling with lower energy.

The criteria for a local minimum with respect to expansion and swap moves are so strong that there are many fewer minima in high-dimensional spaces compared to standard moves. In the terminology in [3], these algorithms are "very large neighborhood search techniques."

In the original work of Boykov et al. [11], the expansion-move algorithm was shown to be applicable to any energy where V_{pq} is a metric, and the swap-move algorithm, to any energy where V_{pq} is a semimetric (that is, a metric except that the triangle inequality may not hold). The work in [31] subsequently relaxed these conditions and showed that the expansion-move algorithm can be used if for all labels α, β, and γ

$$V_{pq}(\alpha,\alpha) + V_{pq}(\beta,\gamma) \leq V_{pq}(\alpha,\gamma) + V_{pq}(\beta,\alpha), \qquad (6)$$

and the swap-move algorithm can be used if for all labels α and β

$$V_{pq}(\alpha,\alpha) + V_{pq}(\beta,\beta) \leq V_{pq}(\alpha,\beta) + V_{pq}(\beta,\alpha). \tag{7}$$

If the energy does not obey these constraints, graph-cut algorithms can still be applied by "truncating" the violating terms [41]. In this case, however, we are no longer guaranteed to find the optimal labeling with respect to expansion or swap moves. In practice, this technique seems to work well only when relatively few terms need to be truncated. For the energy functions used in our benchmarks, only the expansion-move algorithm sometimes requires truncation. We discuss this in more detail at the end of Section 5. The main computational cost of graph cuts lies in computing the minimum cut, which is done via max flow. Our implementation of graph cuts uses the max-flow algorithm in [9], as implemented by the authors. This algorithm is specifically designed for the graphs that arise in vision applications and is shown in [9] to perform particularly well for such graphs. Another potential speedup comes from the fact that many different max-flow computations are run, but for some graph-cut algorithms (in particular, expansion moves), the graph itself does not change very much between iterations. The technique in [25], which often allows results from previous max flows to be reused, could provide an additional speedup but is not part of our current implementation.

3.3 Max-Product Loopy Belief Propagation (LBP)

To evaluate the performance of LBP, we implemented the max-product LBP version, which is designed to find the lowest energy solution. The other main variant of LBP, the sum-product algorithm, does not directly search for a minimum-energy solution but instead computes the marginal probability distribution of each node in the graph. The belief-propagation (BP) algorithm was originally designed for graphs without cycles [38], in which case it produces the exact result for our energy. However, there is nothing in the formulation of BP that prevents it from being tried on graphs with loops. Indeed, BP has been successfully applied to loopy graphs in quite different problem domains such as early vision [17] and error-correcting codes [18]. Detailed descriptions of the LBP algorithm can be found in [16] and [17].

In general, LPB is not guaranteed to converge and may go into an infinite loop switching between two labelings. However, if LBP converges and there are no ties in the min-marginals for nodes, it has a strong local minimum property that is somewhat analogous to that of graph cuts [47], [49].

Felzenszwalb and Huttenlocher [16] present a number of ways to speed up the basic LBP algorithm. In particular, we use the distance transform method described in [16] (when applicable, that is, when the label set is ordered), which significantly reduces the runtime of the algorithm.

We implemented two different variants of LBP: BP-M, an updated version of the max-product LBP implementation of [45], and BP-S, an LBP implementation derived from the TRW-S implementation described in Section 3.4. The most significant difference between the two implementations is in the schedules for passing messages on grids. In the BP-M implementation, messages are passed along rows and then along columns. When a row or column is processed, the algorithm starts at the first node and passes messages in one direction—similar to the forward-backward algorithm for Hidden Markov Models. Once the algorithm reaches the end of a row or column, messages are passed backward along the same row or column. In the BP-S implementation, the nodes are processed in scan-line order, with a forward and backward pass. In the forward pass, each node sends messages to its right and bottom neighbors. In the backward pass, messages are sent to the left and upper neighbors. Another difference between our LBP implementations is how the labeling is computed. In BP-M, each pixel chooses independently the label with the highest belief, whereas in BP-S, the labeling is computed from messages, as described in Section 3.4. Based on some experiments, we do not believe that the performance of BP-S would be improved by adopting the label computing technique of BP-M. Note that for BP-S, we used integers for messages to provide additional efficiency.

The performance of the two versions differs by a surprisingly large margin. In Section 5, we show experimental results of both implementations. On our last benchmark, we also include the BP implementation in [16] in the comparison.

3.4 Tree-Reweighted Message Passing (TRW)

Tree-reweighted message passing [48] is a message-passing algorithm similar, on the surface, to LBP. Let $M^t_{p\to q}$ be the message that pixel p sends to its neighbor q at iteration t; this is a vector of size m (the number of labels). The message update rule is

$$\begin{aligned} M^t_{p\to q}(l_q) = \min_{l_p} \Bigg(c_{pq} \Bigg(d_p(l_p) + \sum_{s\in\mathcal{N}(p)} M^{t-1}_{s\to p}(l_p) \Bigg) \\ - M^{t-1}_{q\to p}(l_p) + V_{pq}(l_p, l_q) \Bigg). \end{aligned} \tag{8}$$

The coefficients c_{pq} are determined in the following way: First, a set of trees from the neighborhood graph (a 2D grid in our case) is chosen so that each edge is in at least one tree. A probability distribution ρ over the set of trees is then chosen. Finally, c_{pq} is set to ρ_{pq}/ρ_p, that is, the probability that a tree chosen randomly under ρ contains edge (p,q) *given* that it contains p. Note that if c_{pq} were set to 1, the update rule would be identical to that of the standard LBP.

An interesting feature of the TRW algorithm is that it computes a lower bound on the energy. We use this lower bound in our experimental results (in Section 5 below) to assess the quality of the solutions. The best solutions are typically within 1 percent of the maximum lower bound.

The original TRW algorithm does not necessarily converge and does not, in fact, guarantee that the lower bound always increases with time. In this paper, we use an improved version of TRW due to [26], which is called sequential TRW, or TRW-S. In this version, the lower bound estimate is guaranteed not to decrease, which results in certain convergence properties. In TRW-S, we first select an arbitrary pixel ordering function $S(p)$. The messages are updated in order of increasing $S(p)$, and at the next iteration, are updated in the reverse order. Trees are constrained to be chains that are monotonic with respect to $S(p)$. Note that the algorithm can be implemented using half as much memory as some versions of BP since it needs to store one message per edge.

Given messages M, we compute labeling l as described in [26]: We go through pixels in the order $S(p)$ and choose label l_p that minimizes

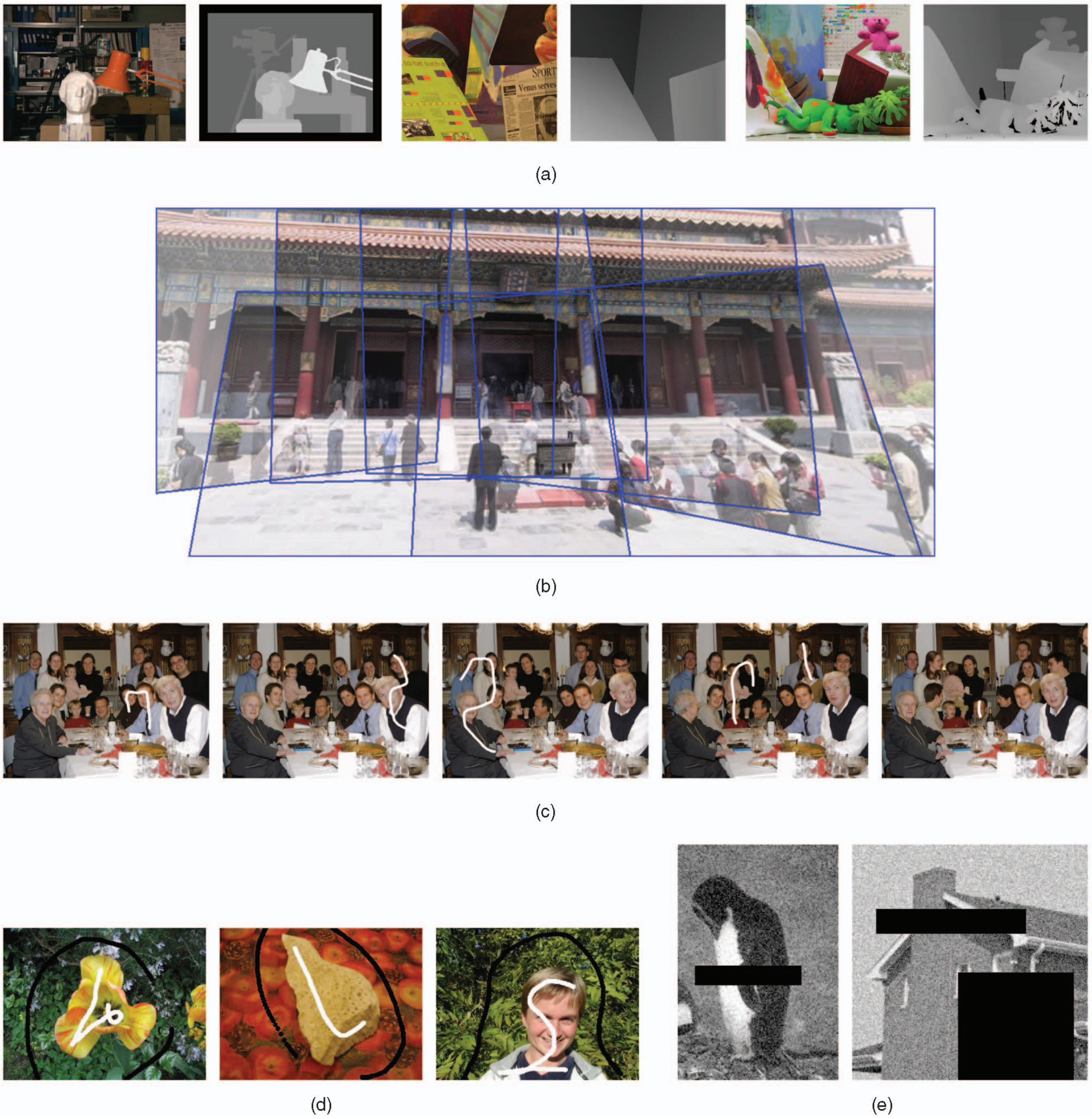


Fig. 2. Images used for our benchmarks. (a) Stereo matching: Tsukuba, Venus, and Teddy left images and true disparities. (b) Photomontage 1: Panorama. (c) Photomontage 2: Family group shot. (d) Binary image segmentation: Flower, Sponge, and Person. (e) Denoising and inpainting: Penguin and House.

$$d_p(l_p) + \sum_{S(q)<S(p)} V_{pq}(l_p, l_q) + \sum_{S(q)>S(p)} M_{q\to p}(l_p).$$

Note that this rule is a heuristic, and there is no guarantee that the energy might not actually increase with time—it is only guaranteed that the lower bound does not decrease. In practice, the energy sometimes starts to oscillate. To deal with this issue, one could keep track of the lowest energy to date and return that state when the algorithm is terminated.

4 Benchmark Problems

For our benchmark problems, we have created a representative set of low-level energy minimization problems drawn from a range of different applications. As with the optimization methods, most of the energy functions and data were contributed by the original authors of the problems. The input images for each benchmark are shown in Fig. 2.

4.1 Stereo Matching

For stereo matching, we follow in the footsteps of Boykov et al. [10] and Tappen and Freeman [45] and use a simple energy function for stereo, applied to images from the widely used Middlebury stereo data set [42] (see Fig. 2a). The labels are the disparities, and the data costs are the absolute color differences between corresponding pixels for each disparity. We use the cost variant by Birchfield and Tomasi [6] for increased robustness to image sampling.

We use different smoothness costs for the different image pairs to make the optimization problems more varied. For "Tsukuba" with $m = 16$ labels, we use a truncated linear cost $(k = 1, V_{\max} = 2)$ with $\lambda = 20$. For "Venus" with $m = 20$ labels, we use a truncated quadratic cost $(k = 2, V_{\max} = 7)$ with $\lambda = 50$. Since this smoothness term is not a metric, applying the expansion-move algorithm requires truncation. For "Teddy" with $m = 60$ labels, we use the Potts model $(k = 1, V_{\max} = 1)$ with $\lambda = 10$. The default local smoothness weight is $w_{pq} = 1$ at all pixels. For "Tsukuba" and "Teddy," we increase the weight at locations where the intensity gradient ∇_{pq} in the left image is small: We use $w_{pq} = 2$ if $|\nabla_{pq}| \leq 8$ for "Tsukuba" and $w_{pq} = 3$ if $|\nabla_{pq}| \leq 10$ for "Teddy."

4.2 Photomontage

The Photomontage system [1] seamlessly stitches together multiple photographs for a variety of photo merging applications. We include benchmarks for two such applications: panoramic stitching and group photo merging (see Figs. 2b and 2c). The input is a set of aligned images $S_1, S_2, \ldots, S_m$ of equal dimension; the labels are the image indices, that is, $1, 2, \ldots, m$; the final output image is formed by copying colors from the input images according to the computed labeling. If two neighbors p and q are assigned the same input image, they should appear natural in the composite, and so, $V_{pq}(i, i) = 0$. If $l_p \neq l_q$, we say that a seam exists between p and q; then, V_{pq} measures how visually noticeable the seam is in the composite. The data term $d_p(i)$ is zero if pixel p is in the field of view of image i and ∞ otherwise.

The first benchmark, "Panorama," stitches together the panorama in Fig. 2b [1, Fig. 8]. The smoothness energy, derived from [34], is

$$V_{pq}(l_p, l_q) = |S_{l_p}(p) - S_{l_q}(p)| + |S_{l_p}(q) - S_{l_q}(q)|. \tag{9}$$

This energy function is suitable for the expansion-move algorithm without truncation.

The second benchmark, "Family," stitches together the five group photographs shown in Fig. 2c [1, Fig. 1]. The best depiction of each person is to be included in a composite. Photomontage itself is interactive, but to make the benchmark repeatable, the user strokes are saved into a data file. For any pixel p underneath a drawn stroke, $d_p(l_p) = 0$ if l_p equals the user-indicated source image and ∞ otherwise. For pixels p not underneath any strokes, $d_p(l_p) = 0$ for all labels. The smoothness terms are modified from the first benchmark to encourage seams along strong edges. More precisely, we divide the right-hand side of (9) by $|\nabla_{pq} S_{l_p}| + |\nabla_{pq} S_{l_q}|$, where $\nabla_{pq} I$ is the gradient between pixels p and q in image I. The expansion-move algorithm is applicable to this energy only after truncating certain terms, but it continues to work well in practice.

4.3 Binary Image Segmentation

Binary MRFs are also widely used in medical image segmentation [8], stereo matching using minimal surfaces [12], [43], and video segmentation using stereo disparity cues [27]. As previously mentioned, for the natural Ising model smoothness cost, the global minimum can be computed rapidly via graph cuts [21]; this result has been generalized to other smoothness costs in [31]. Nevertheless, such energy functions still form an interesting benchmark, since there may well be other heuristic algorithms that perform faster while achieving nearly the same level of performance.

Our benchmark consists of three segmentation problems, inspired by the interactive segmentation algorithm in [8] and its more recent extensions [39]. As with our Photomontage stitching example, this application requires user interaction; we handle this issue as above, by saving the user interactions to a file and using them to derive the data costs. Fig. 2d shows the three input images and the user strokes.

The data cost is the log likelihood of a pixel belonging to either the foreground or background and is modeled as two separate Gaussian mixture models, as in [39]. The smoothness term is a standard Potts model that is contrast sensitive

$$w_{pq} = \exp(-\beta \|S(p) - S(q)\|^2) + \lambda_2, \tag{10}$$

where $\lambda = 50$, $\lambda_2 = 10$, and $S(p)$ and $S(q)$ are the RGB colors of two neighboring pixels p and q. The quantity β is set to $(2\langle \|S(p) - S(q)\|^2 \rangle)^{-1}$, where the expectation denotes an average over the image, as motivated in [39]. The purpose of λ_2 is to remove small and isolated areas that have high contrast.

4.4 Image Denoising and Inpainting

For the denoising and inpainting benchmark, we use the "Penguin" image, which appears in [16, Fig. 8], and the "House" image, a standard test image in the denoising literature. We added random noise to each pixel and also obscured a portion of each image (see Fig. 2e). The labels are intensities $(m = 256)$, and the data cost d_p for each pixel is the squared difference between the label and the observed intensity, except in the obscured portions, where $d_p(l_p) = 0$ for all intensities. For the "Penguin" image, we use a truncated quadratic smoothness cost $(k = 2, V_{\max} = 200)$ with $\lambda = 25$. For the "House" image, we use a nontruncated quadratic cost $(k = 2, V_{\max} = \infty)$ with $\lambda = 5$. In both cases, the expansion-move algorithm requires truncation. Unlike all of the other benchmarks, the "House" example is a convex minimization problem amenable to quadratic solvers, since both data and smoothness costs are quadratics. The implications of this are discussed in Section 5.

5 Experimental Results

The experimental results from running the different optimization algorithms on these benchmarks are given in Figs. 3 (stereo), 4 (Photomontage), 5 (binary image segmentation), and 6 (denoising and inpainting). The six methods we compare are ICM, BP-M (max-product LBP), BP-S (LBP implementation derived from TRW-S), Swap (graph cuts using swap moves), Expansion (graph cuts using expansion moves), and TRW-S. The x-axis of the plots in Figs. 3, 4, 5, and 6 shows runtimes in seconds on a log scale. The y-axis shows the energy of the different methods over time. As mentioned, instead of showing absolute energy values, we make use of TRW's ability to compute a lower bound on the energy of the optimal solution and normalize the energy by dividing it by the best lower bound computed by TRW-S. The lower bound increases monotonically [26], and the computed lower-bound values are included in the plots.

On all benchmarks, the best methods achieve results that are extremely close to the global minimum, with less than 1 percent error in all cases, and often less than 0.1 percent. For example, on "Tsukuba," TRW-S gets to within 0.02 percent of

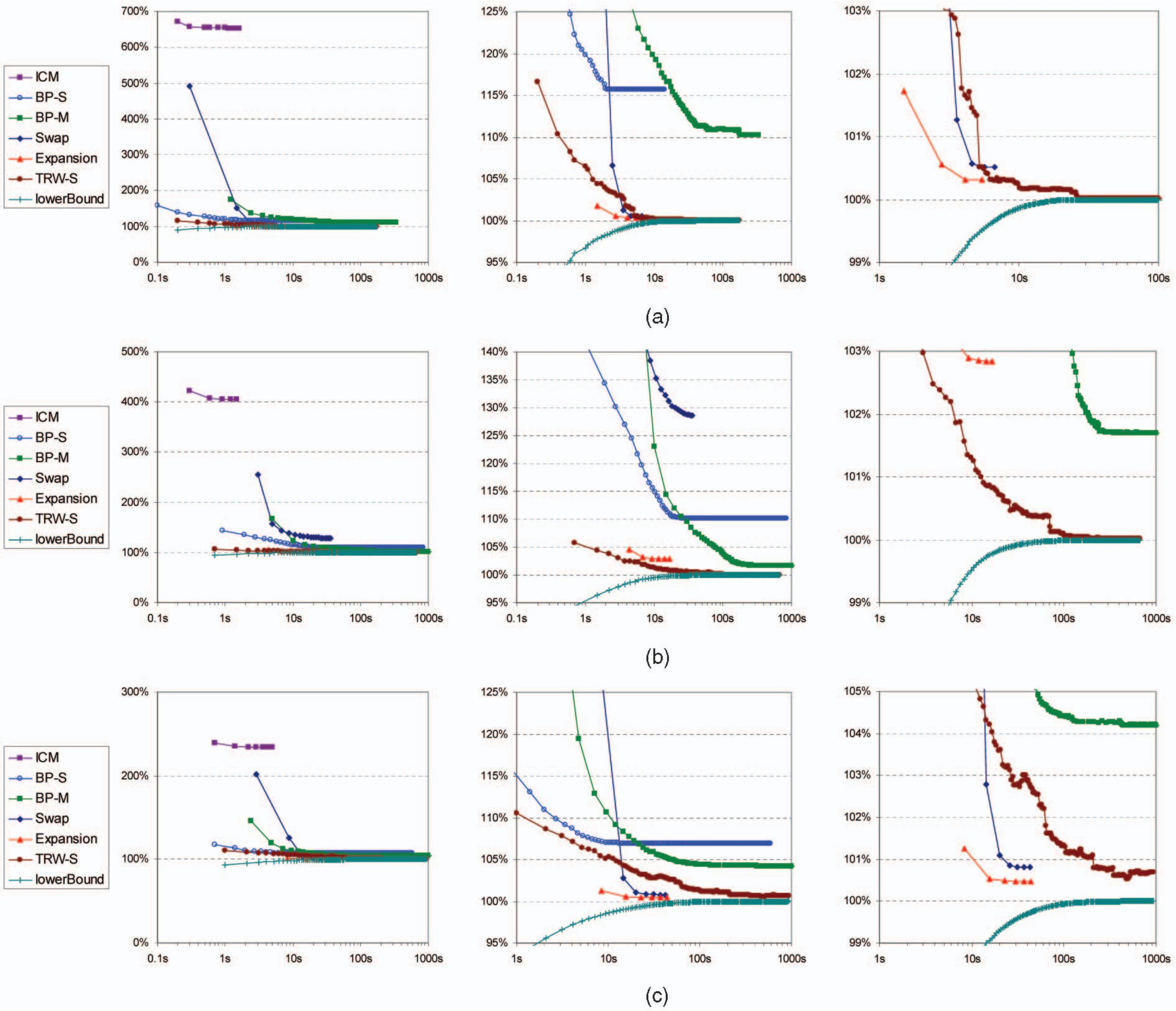


Fig. 3. Results on the stereo matching benchmarks. (a) "Tsukuba" energy with truncated linear smoothness cost V. (b) "Venus" energy with truncated quadratic smoothness cost V. (c) "Teddy" energy with the Potts model for V. All plots show the runtime on the x-axis using a log scale. The y-axis shows the energy relative to the maximum lower bound achieved by the TRW-S algorithm. Each row contains multiple plots of the same curves, with increasing zoom from left to right. The zoomed plots may not contain the poorer performing algorithms. The plots in Figs. 4, 5, and 6 are generated in the same manner.

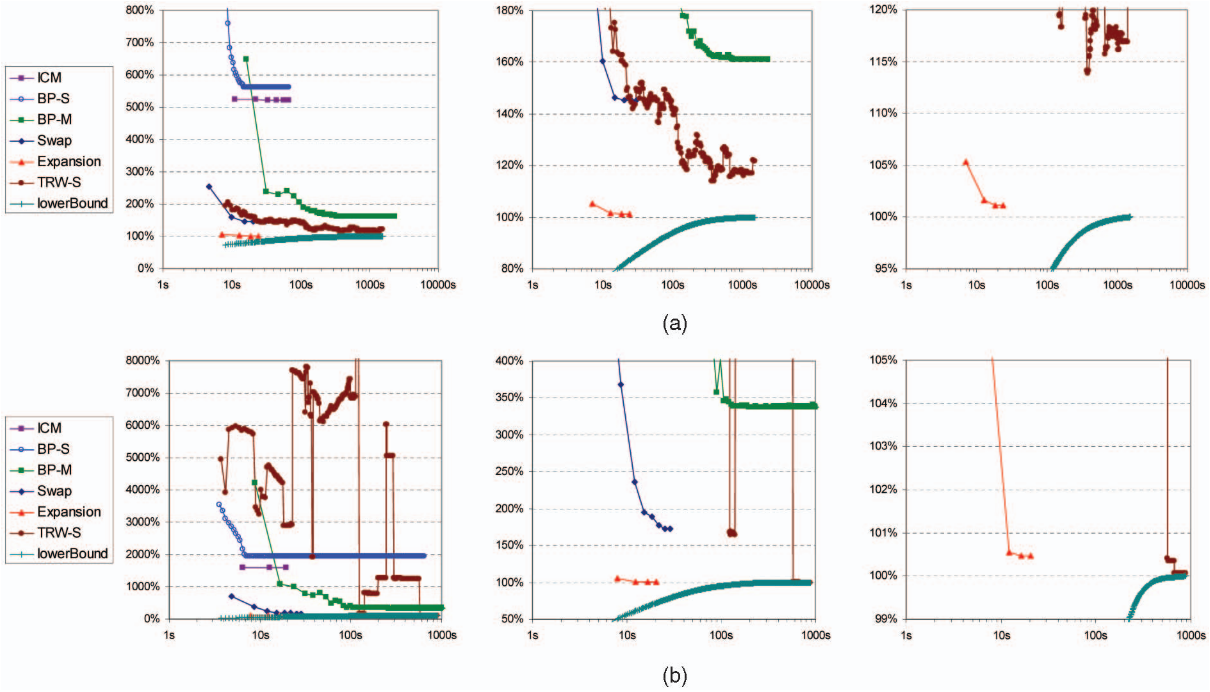
the optimum, whereas on "Panorama," Expansion is within 0.9 percent. These statistics may actually slightly understate the performance of the methods since they are based on the TRW-S lower bound rather than the global minimum, which is unknown.

The individual plots show some interesting features. On the stereo benchmarks (Fig. 3), Expansion finds near-optimal solutions quickly and is the overall winner for "Teddy." Although slower, TRW-S does extremely well, eventually beating all other methods on "Tsukuba" and "Venus." Swap is slower and performs slightly worse than Expansion in all cases. The "Venus" results are particularly interesting: Expansion does much worse here than on the other stereo images, presumably due to the quadratic smoothness term, and Swap does even worse. There is also a large gap between the performance of the two LBP implementations, and BP-M does quite well (second only to TRW-S).

The Photomontage benchmarks (Fig. 4), with their label-dependent smoothness costs, seem to present the largest challenge for the energy minimization methods, many of which come nowhere near the best solution. The exception is Expansion, which finds solutions with less than 1 percent error on both benchmarks in a few iterations. TRW-S oscillates wildly but eventually beats Expansion on "Family" though not on "Panorama." As noted earlier, if TRW-S was used in an application with limited time, it would be necessary to keep track of the current best solution.

On the binary image segmentation benchmarks (Fig. 5), graph cuts are guaranteed to compute the global minimum, as is TRW-S (but not the original TRW [48]). Both LBP implementations come extremely close (under 0.1 percent error in all cases) but never actually attain the global minimum.

Our study has focused on classical nonconvex energy minimization problems that arise from early vision problems such as stereo. The much simpler problem of convex energy minimization is also worth studying. In our final benchmark, denoising and inpainting (Fig. 6), we experiment both with a nonconvex (truncated quadratic) V for the "Penguin" data set and a convex (quadratic) V for the "House" data set. Since in the latter case, both data and smoothness energies are quadratic, this is a Gaussian MRF for which a real-valued



(a)

(b)

Fig. 4. Results on the Photomontage benchmarks. V measures how noticeable a seam is in the final composite and depends both on labels and intensities. (a) "Panorama." (b) "Family."

solution can be found in closed form. On this problem, hierarchically preconditioned conjugate gradient descent [44], labeled HBF in Fig. 6b, outperforms all other algorithms by a large margin, requiring only five iterations and a fraction of a second. However, since the resulting floating-point solution is rounded to integers, the resulting energy is slightly higher than the integer solution found by TRW-S. BP requires a dozen or more iterations and takes over 10 seconds to converge, whereas the graph-cut variants do much worse and are even slower, perhaps due to the nonmetric smoothness terms and also since there are no constant-valued regions to propagate over large distances. On these benchmarks, we also include results for the popular LBP implementation in [16], which is labeled BP-P in the plots. On "Penguin," it performs comparable to our two BP implementations, whereas on "House," it converges faster and to a slightly lower energy. However, there is also a small increase in energy later on. Since BP-P does not implement our API and requires constant w_{pq}, we did not run it on our other benchmarks.

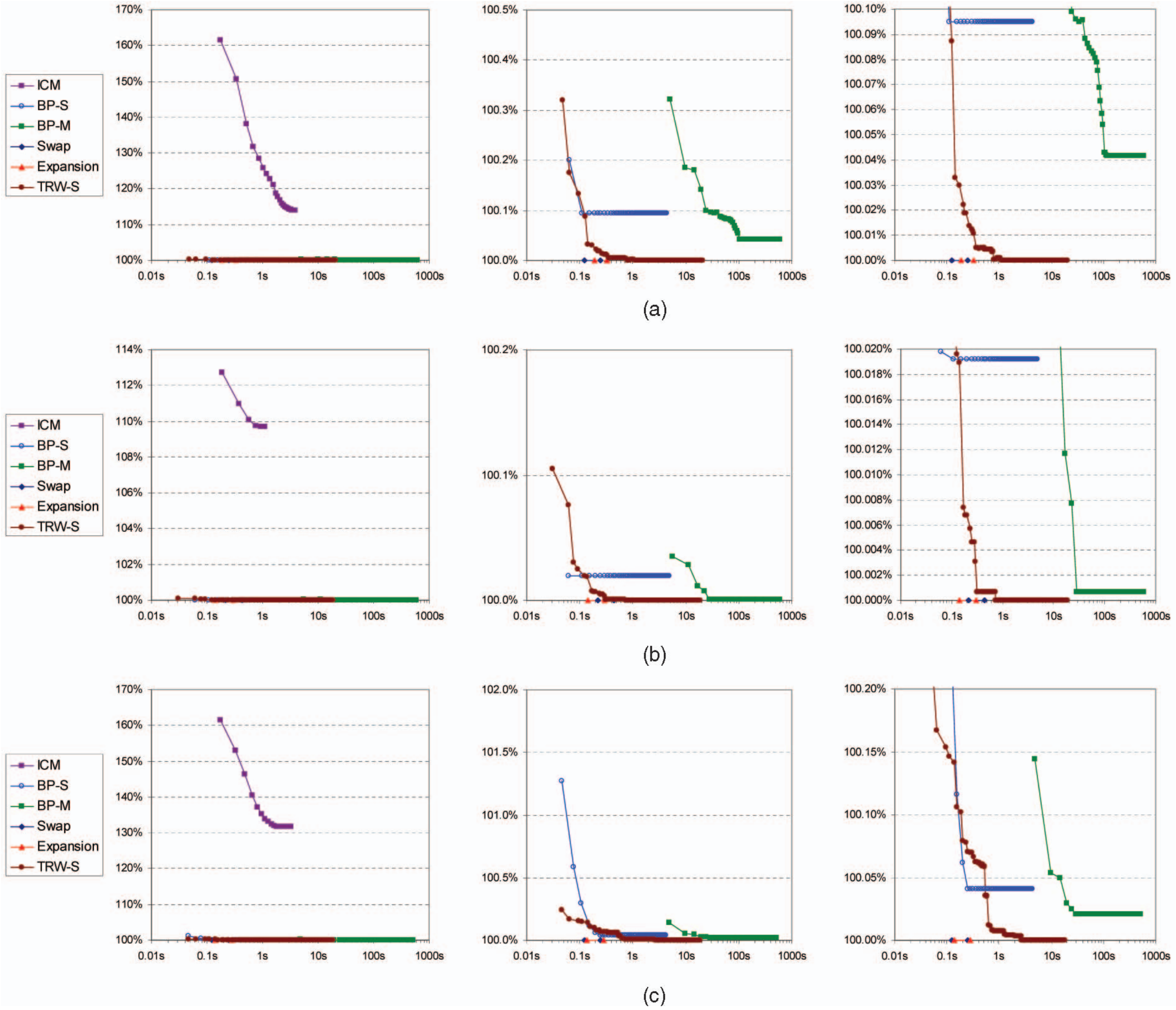
Comparing the results of the quadratic energy (Fig. 6b) with those of the nonquadratic energy (Fig. 6a), we see that the results are fairly consistent, except that we now have a much faster algorithm (HBF) that does even better. It would be interesting to investigate whether a continuation method [7] used in conjunction with conjugate gradient might perform well on nonquadratic but (piecewise) smoothly varying MRFs. It is also possible that techniques that combine graph cuts with expectation-maximization, such as [36], would be effective on such problems, since they provide high-quality stereo results on images without regions of constant disparity.

Images showing the resulting labelings for all methods on all benchmarks are provided on the project Web page at http://vision.middlebury.edu/MRF/, where they can be compared interactively. In terms of visual quality of the resulting labelings, the ICM results look noticeably worse, but the others are difficult to distinguish on most of our benchmarks. The major exception is the Photomontage benchmarks, for which we include the resulting labeling images in this paper. On "Panorama," shown in Fig. 7, ICM, BP-S, and Swap all make some major errors, leaving slices of people floating in the air. BP-M does only slightly better, whereas both Expansion and TRW-S do quite well, with TRW-S producing the fewest noticeable seams. Similarly, on "Family," shown in Fig. 8, ICM and BP-S make major errors, Swap and BP-M do slightly better but still produce noticeable errors, whereas again Expansion and TRW-S both work very well, with nearly identical results.

A final issue deserving investigation is the impact of truncation when using the expansion-move algorithm. Truncation is required when the so-called regularity condition (6) on the smoothness term V is not met. This is the case for the "Family" Photomontage benchmark, which uses a contrast-sensitive smoothness term, and the "Venus" stereo and "Penguin" and "House" denoising benchmarks, which use (truncated and nontruncated) quadratic smoothness costs. However, we found that the total number of terms that need to be truncated is very low—typically a fraction of 1 percent. The precise numbers for our benchmarks are 0.002 percent for "Venus," 0.009 percent for "Family," 0.03 percent for "Penguin," and 0.5 percent for "House." Thus, it is unlikely that truncation strongly affects the performance of the expansion-move algorithm in our benchmarks. Furthermore, the QBPO algorithm described in [29] can often solve problems that violate the regularity condition.

6 Discussion

The strongest impression that one gets from our data is of how much better modern energy minimization methods are than ICM and how close they come to computing the global minimum. We do not believe that this is purely due to flaws



(a)

(b)

(c)

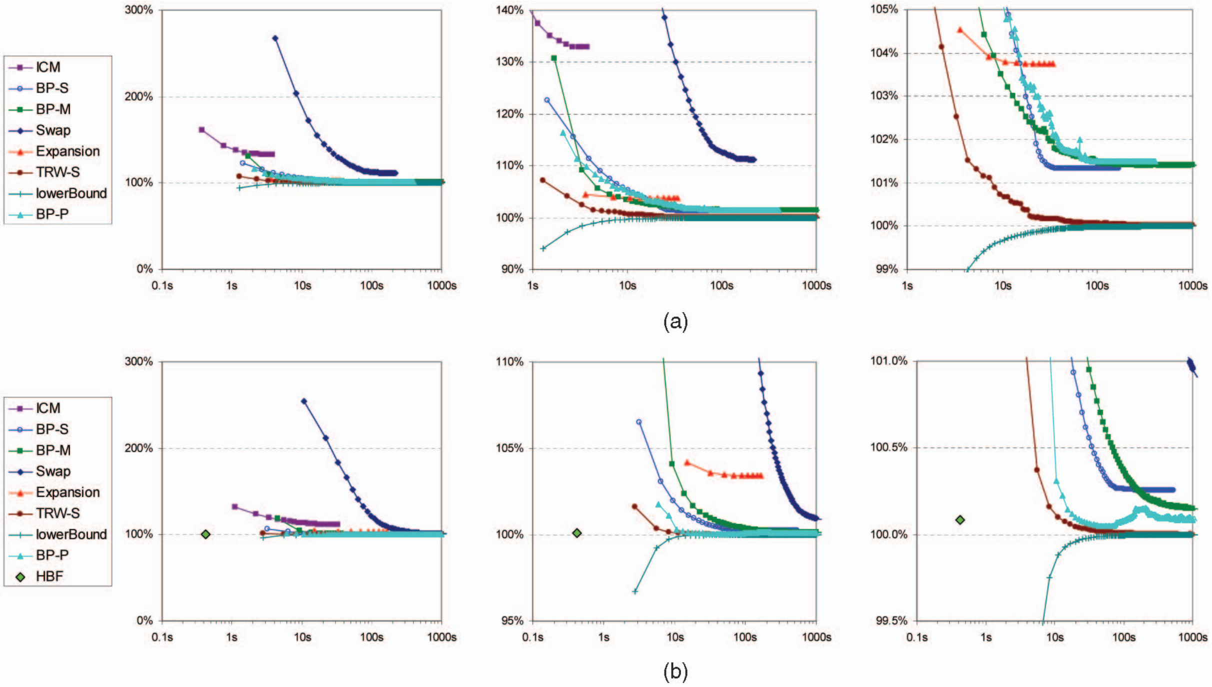
Fig. 5. Results on the binary image segmentation benchmarks. Here, the global minimum can be computed rapidly by graph cuts or TRW-S and is used to normalize the energies. (a) "Flower." (b) "Sponge." (c) "Person."

in ICM but simply reflects the fact that the methods used until the late 1990s performed poorly. (As additional evidence, Boykov et al. [11] compared the energy produced by graph cuts with simulated annealing and obtained a similarly large improvement.) We believe that our study demonstrates that the state of the art in energy minimization has advanced significantly in the last few years.

There is also a dramatic difference in performance among the different energy minimization methods on our benchmarks, and on some of the benchmarks, there are clear winners. On the Photomontage benchmark, Expansion performs best, which provides some justification for the fact that this algorithm is used by various image stitching applications [1], [2]. On the stereo benchmark, the two best methods seem to be TRW-S and Expansion. There are also some obvious paired comparisons; for instance, there never seems to be any reason to use Swap instead of Expansion. In terms of runtime, Expansion is clearly the winner among the competitive methods (that is, all except ICM), but it should be noted that not all methods have been optimized for speed equally.

There is clearly a need for more research on message-passing methods such as TRW-S and LBP. Although LBP is a well-regarded and widely used method, both of our LBP implementations perform surprisingly poorly on many of our benchmarks. TRW-S, which has not been widely used in vision, gives consistently strong results. In addition, the lower bound on the energy provided by TRW-S proved extremely useful in our study. For a user of energy minimization methods, this lower bound can serve as a confidence measure, providing assurance that the solution obtained has near-optimal energy. Another area that needs investigation is the use of graph-cut algorithms for wider classes of energy functions than the limited ones they were originally designed for. It is interesting that the benchmarks that are most challenging for the expansion and swap-move algorithms (such as "Venus" and the denoising examples) use a V that is not a metric. As mentioned above, however, only the expansion-move algorithm requires truncation (and only on a small percentage of terms), so truncation does not appear to be the main issue. It is worth investigating whether some very recent algorithms based on graph cuts, such the range-move algorithm in [46] or the primal-dual method in [32], would give better results on these benchmarks.

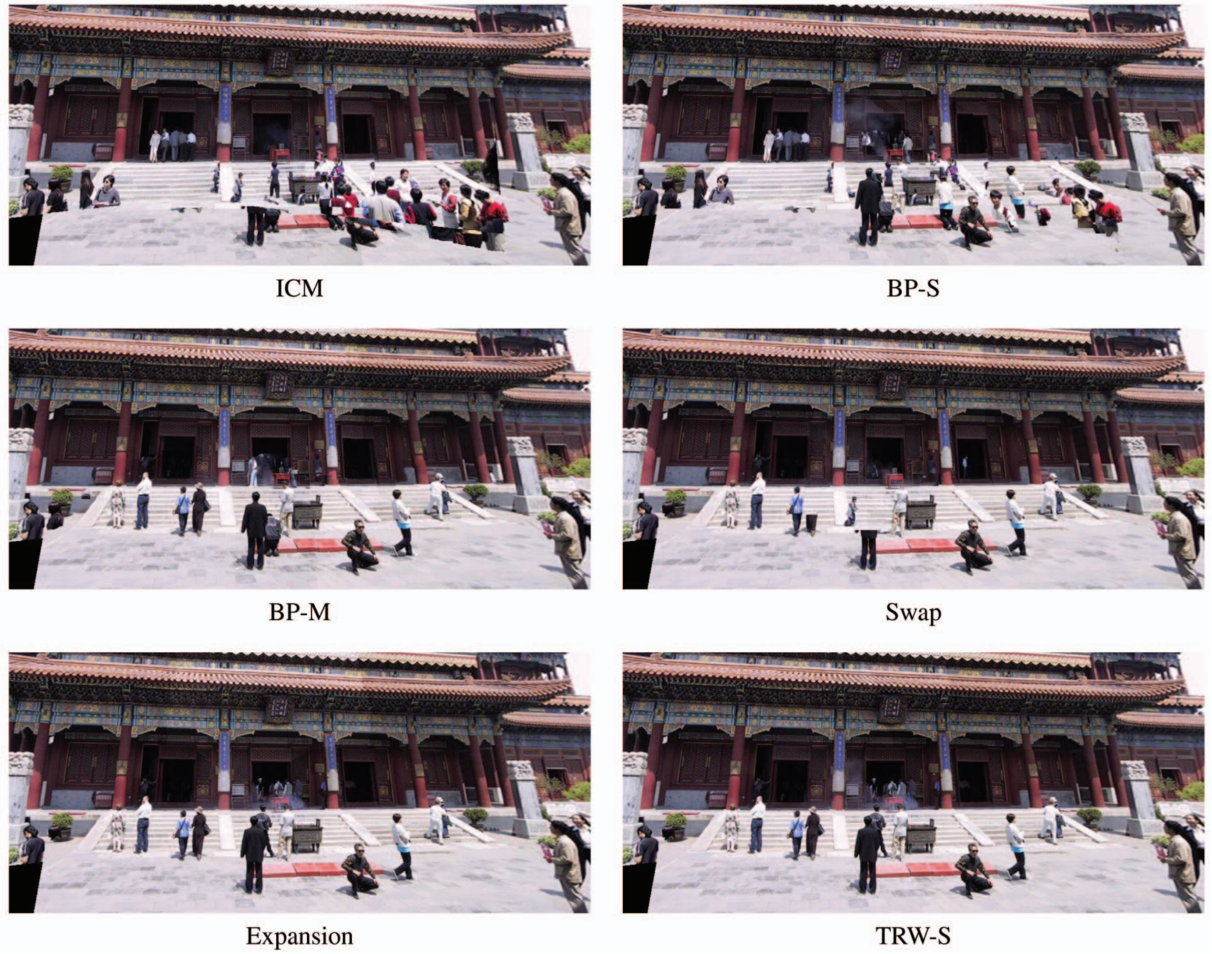
Another important issue centers around the use of energy to compare energy minimization algorithms. The goal in computer vision, of course, is not to compute the lowest energy solution to a problem but rather the most accurate one. Although computing the global minimum was shown to be



(a)

(b)

Fig. 6. Results on the denoising and inpainting benchmarks. (a) “Penguin” energy, with truncated quadratic smoothness cost V. (b) “House” energy, with nontruncated quadratic smoothness cost V. On these benchmarks, we include the LBP implementation in [16], which is labeled BP-P. The “House” example is a convex energy minimization problem amenable to quadratic solvers; we include the results of HBF, the fast algorithm in [44].


ICM

BP-S

BP-M

Swap

Expansion

TRW-S

Fig. 7. The resulting images for the “Panorama” Photomontage benchmark.

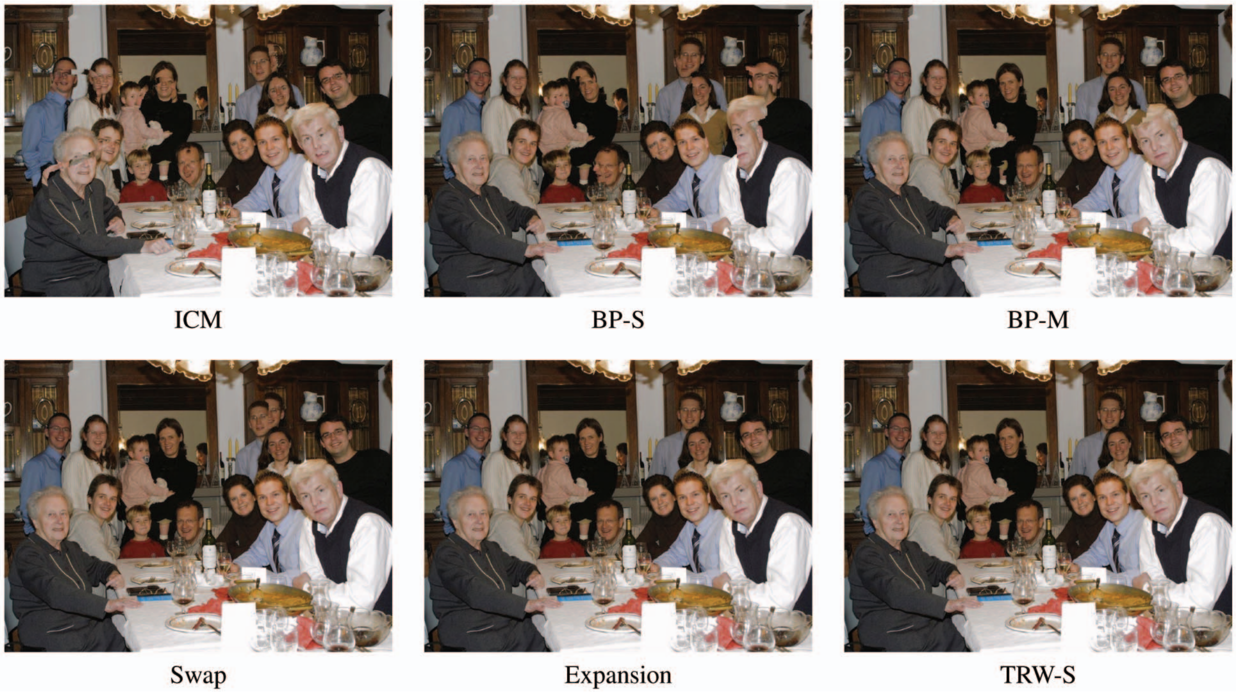


Fig. 8. The resulting images for the “Family” Photomontage benchmark.

NP-hard [11], it is sometimes possible for special cases. For example, the energy minimization problem can be recast as an integer program, which can be solved as a linear program; if the linear program’s solutions happen to be integers, they are the global minimum. This is the basis for the approach taken by Meltzer et al. [37], who demonstrated that the global minimum could be computed for several common energy functions on the Middlebury images. The global minimum has only a slightly lower energy than that produced by graph cuts or LBP. In addition, Meltzer et al. [37] point out that the global minimum is no more accurate than the results achieved with graph cuts or LBP. More precisely, according to Meltzer et al. [37], at best, graph cuts produces an energy that is 0.018 percent over the global minimum, whereas at worst, the energy is 3.6 percent larger; at best, LBP gives an energy that is 3.4 percent higher, and at worst, 30 percent.

In light of these results, it is clear that for the models we have considered, better minimization techniques are unlikely to produce significantly more accurate labelings. For the Middlebury stereo benchmarks, this is particularly clear: the best methods produce energies that are extremely close to the global minimum; the global minimum, when known, is no more accurate; and, in fact, the ground truth has substantially higher energy. To illustrate this point, we compare the ground-truth energies with those computed by the minimization techniques in Table 1. Note that since our MRF model does not account for occlusions, for fairness, only the energies of the nonoccluded regions (shown in the right columns in the table) should be considered. Even so, all of the competitive methods easily beat the energy corresponding to the ground truth, indicating the limitations of the simple MRF stereo model used.

However, it is still important to compare energy minimization algorithms using the energy they produce as a benchmark. Creating more accurate models will not lead to better results if good labelings under these models cannot be found. It is also difficult to gauge the power of a model without the ability to produce low-energy labelings.

7 Conclusion

In this paper, we introduced a set of energy minimization benchmarks drawn from published energy functions used for stereo, image stitching, interactive segmentation, and denoising. We used these benchmarks to compare the solution quality and runtime of several common energy minimization algorithms. We are making our minimization code available to other researchers by providing a general-purpose software interface that allows easy switching between optimization methods. We also provide the code for all of the benchmarks.

Although we have investigated quite a few benchmarks and algorithms, we expect that additional ones will be added to the project page over time. (See, for example, three

TABLE 1
Energy of Ground Truth versus Minimization Techniques on the Stereo Benchmarks

	Tsukuba		Venus		Teddy	
	all	nonocc	all	nonocc	all	nonocc
Ground Truth	219%	125%	144%	113%	170%	145%
ICM	683%	861%	405%	495%	233%	301%
Expansion	100%	100%	103%	103%	100%	100%
Swap	100%	100%	129%	136%	100%	100%
TRW-S	100%	100%	100%	100%	100%	100%
BP-S	112%	115%	110%	114%	106%	110%
BP-M	111%	113%	102%	103%	104%	106%

Energies are given relative to the minimum, rounded to the nearest percent. They are computed both everywhere (“all”), and in nonoccluded areas only (“nonocc”). The latter numbers provide a better assessment of ground-truth energies since our MRF model does not account for occlusions. Still, even when only considering nonoccluded areas, most minimization techniques achieve energies that are significantly lower than the energy of the ground truth.

relevant papers presented in the 2007 IEEE CS Conference on Computer Vision and Pattern Recognition (CVPR), [33], [40], [46].) In terms of benchmarks, it would also be interesting to investigate different grid topologies (such as the 8-connected topology for 2D or 26-connected topology for 3D), as well as nonlocal topologies such as those used with multiple depth maps [28].

There are many algorithms that could be naturally incorporated into our study. Two algorithms that appear particularly interesting are the TRW-based method in [37], which could potentially achieve the global minimum on some of our benchmarks, and the graph-cut algorithm in [32], which generalize the expansion-move algorithm so that (like TRW) it also computes a lower bound on the energy. The area of multiresolution techniques also bears substantial promise, as shown in [2].

ACKNOWLEDGMENTS

The original suggestion for a common API was made by Eero Simoncelli.

REFERENCES

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive Digital Photomontage," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 294-302, 2004.

[2] A. Agarwala, C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski, "Panoramic Video Textures," *ACM Trans. Graphics,* vol. 24, no. 3, pp. 821-827, 2005.

[3] R. Ahuja, Ö. Ergun, J. Orlin, and A. Punnen, "A Survey of Very Large-Scale Neighborhood Search Techniques," *Discrete Applied Math.,* vol. 123, nos. 1-3, pp. 75-102, 2002.

[4] S. Barnard, "Stochastic Stereo Matching over Scale," *Int'l J. Computer Vision,* vol. 3, no. 1, pp. 17-32, 1989.

[5] J. Besag, "On the Statistical Analysis of Dirty Pictures (with discussion)," *J. Royal Statistical Soc., Series B,* vol. 48, no. 3, pp. 259-302, 1986.

[6] S. Birchfield and C. Tomasi, "A Pixel Dissimilarity Measure That Is Insensitive to Image Sampling," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 4, pp. 401-406, Apr. 1998.

[7] A. Blake and A. Zisserman, *Visual Reconstruction.* MIT Press, 1987.

[8] Y. Boykov and M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images," *Proc. Eighth IEEE Int'l Conf. Computer Vision,* vol. 1, pp. 105-112, 2001.

[9] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1124-1137, Sept. 2004.

[10] Y. Boykov, O. Veksler, and R. Zabih, "Markov Random Fields with Efficient Approximations," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 648-655, 1998.

[11] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 11, pp. 1222-1239, Nov. 2001.

[12] C. Buehler, S. Gortler, M. Cohen, and L. McMillan, "Minimal Surfaces for Stereo Vision," *Proc. Seventh European Conf. Computer Vision,* vol. 3, pp. 885-899, 2002.

[13] P. Chou and C. Brown, "The Theory and Practice of Bayesian Image Labeling," *Int'l J. Computer Vision,* vol. 4, no. 3, pp. 185-210, 1990.

[14] K. Driesen and U. Hölzle, "The Direct Cost of Virtual Function Calls in C++," *Proc. ACM SIGPLAN Conf. Object-Oriented Programming Systems, Languages, and Applications,* pp. 306-323, 1996.

[15] A. Efros and W. Freeman, "Image Quilting for Texture Synthesis and Transfer," *Proc. ACM SIGGRAPH '01,* pp. 341-346, 2001.

[16] P. Felzenszwalb and D. Huttenlocher, "Efficient Belief Propagation for Early Vision," *Int'l J. Computer Vision,* vol. 70, no. 1, pp. 41-54, 2006.

[17] W. Freeman, E. Pasztor, and O. Carmichael, "Learning Low-Level Vision," *Int'l J. Computer Vision,* vol. 40, no. 1, pp. 25-47, 2000.

[18] B. Frey and D. MacKay, "A Revolution: Belief Propagation in Graphs with Cycles," *Advances in Neural Information Processing Systems,* 1997.

[19] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, pp. 721-741, 1984.

[20] S. Geman and C. Graffigne, "Markov Random Field Image Models and Their Applications to Computer Vision," *Proc. Int'l Congress of Math.,* pp. 1496-1517, 1986.

[21] D. Greig, B. Porteous, and A. Seheult, "Exact Maximum A Posteriori Estimation for Binary Images," *J. Royal Statistical Soc., Series B,* vol. 51, no. 2, pp. 271-279, 1989.

[22] P. Hammer, "Some Network Flow Problems Solved with Pseudo-Boolean Programming," *Operations Research,* vol. 13, pp. 388-399, 1965.

[23] D. Hochbaum, "An Efficient Algorithm for Image Segmentation, Markov Random Fields and Related Problems," *J. ACM,* vol. 48, no. 4, pp. 686-701, 2001.

[24] H. Ishikawa, "Exact Optimization for Markov Random Fields with Convex Priors," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 10, pp. 1333-1336, Oct. 2003.

[25] P. Kohli and P. Torr, "Dynamic Graph Cuts for Efficient Inference in Markov Random Fields," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 12, pp. 2079-2088, Dec. 2007.

[26] V. Kolmogorov, "Convergent Tree-Reweighted Message Passing for Energy Minimization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 10, pp. 1568-1583, Oct. 2006.

[27] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother, "Probabilistic Fusion of Stereo with Color and Contrast for Bilayer Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 9, pp. 1480-1492, Sept. 2006.

[28] V. Kolmogorov and C. Rother, "Comparison of Energy Minimization Algorithms for Highly Connected Graphs," *Proc. Ninth European Conf. Computer Vision,* pp. 1-15, 2006.

[29] V. Kolmogorov and C. Rother, "Minimizing Nonsubmodular Functions with Graph Cuts—A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 7, pp. 1274-1279, 2007.

[30] V. Kolmogorov and R. Zabih, "Multi-Camera Scene Reconstruction via Graph Cuts," *Proc. Seventh European Conf. Computer Vision,* pp. 82-96, 2002.

[31] V. Kolmogorov and R. Zabih, "What Energy Functions can be Minimized via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 2, pp. 147-159, Feb. 2004.

[32] N. Komodakis and G. Tziritas, "Approximate Labeling via Graph Cuts Based on Linear Programming," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 8, pp. 1436-1453, Aug. 2007.

[33] N. Komodakis, G. Tziritas, and N. Paragios, "Fast, Approximately Optimal Solutions for Single and Dynamic MRFs," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* June 2007.

[34] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts," *ACM Trans. Graphics,* pp. 277-286, 2003.

[35] S. Li, *Markov Random Field Modeling in Computer Vision.* Springer, 1995.

[36] M. Lin and C. Tomasi, "Surfaces with Occlusions from Layered Stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 8, pp. 1073-1078, Aug. 2004.

[37] T. Meltzer, C. Yanover, and Y. Weiss, "Globally Optimal Solutions for Energy Minimization in Stereo Vision Using Reweighted Belief Propagation," *Proc. 11th IEEE Int'l Conf. Computer Vision,* pp. 428-435, 2005.

[38] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[39] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut'—Interactive Foreground Extraction Using Iterated Graph Cuts," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 309-314, 2004.

[40] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing Binary MRFs via Extended Roof Duality," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* June 2007.

[41] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake, "Digital Tapestry," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 589-596, 2005.

[42] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *Int'l J. Computer Vision,* vol. 47, nos. 1-3, pp. 7-42, 2002.

[43] D. Snow, P. Viola, and R. Zabih, "Exact Voxel Occupancy with Graph Cuts," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 345-352, 2000.

[44] R. Szeliski, "Locally Adapted Hierarchical Basis Preconditioning," *ACM Trans. Graphics,* vol. 25, no. 3, pp. 1135-1143, 2006.

[45] M. Tappen and W. Freeman, "Comparison of Graph Cuts with Belief Propagation for Stereo, Using Identical MRF Parameters," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* pp. 900-907, 2003.

[46] O. Veksler, "Graph Cut Based Optimization for MRFs with Truncated Convex Priors," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2007.

[47] M. Wainwright, T. Jaakkola, and A. Willsky, "Tree Consistency and Bounds on the Performance of the Max-Product Algorithm and Its Generalizations," *Statistics and Computing,* vol. 14, pp. 143-166, Apr. 2004.

[48] M. Wainwright, T. Jaakkola, and A. Willsky, "Map Estimation via Agreement on Trees: Message-Passing and Linear Programming," *IEEE Trans. Information Theory,* vol. 51, no. 11, pp. 3697-3717, 2005.

[49] Y. Weiss and W. Freeman, "On the Optimality of Solutions of the Max-Product Belief Propagation Algorithm in Arbitrary Graphs," *IEEE Trans. Information Theory,* vol. 47, no. 2, pp. 723-735, 2001.

[50] J. Yedidia, W. Freeman, and Y. Weiss, "Generalized Belief Propagation," *Advances in Neural Information Processing Systems,* pp. 689-695, 2000.

Richard Szeliski received the PhD degree in computer science from Carnegie Mellon University, Pittsburgh, in 1988. He joined Microsoft Research in 1995. He leads the Interactive Visual Media Group at Microsoft Research, which does research in digital and computational photography, video scene analysis, 3D computer vision, and image-based rendering. Prior to joining Microsoft Research, he worked at Bell-Northern Research, Schlumberger Palo Alto Research, the Artificial Intelligence Center of SRI International, and the Cambridge Research Laboratory of Digital Equipment Corp. He has published more than 125 research papers in computer vision, computer graphics, medical imaging, and neural nets, as well as the book *Bayesian Modeling of Uncertainty in Low-Level Vision*. He was a program committee chair of the Eighth International Conference on Computer Vision (ICCV '01) and the 1999 Vision Algorithms Workshop, served as an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and on the editorial board of the *International Journal of Computer Vision*, and is a founding editor of *Foundations and Trends in Computer Graphics and Vision*. He is a fellow of the IEEE and the IEEE Computer Society.

Ramin Zabih received the PhD degree in computer science from Stanford University after graduating from the Massachusetts Institute of Technology. He is a professor of computer science at Cornell University, where he has taught since 1994. Since 2000, he has also held a joint appointment in the Radiology Department, Weill Cornell Medical College of Cornell University, New York. His research interests lie in early vision and its applications, especially for medical imaging, and in the use of combinatorial optimization techniques. Since 2005, he has been an associate editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence*, and in 2007, he served as a program cochair for the IEEE CS Conference on Computer Vision and Pattern Recognition (CVPR). He is a senior member of the IEEE and the IEEE Computer Society.

Daniel Scharstein studied computer science at the Universität Karlsruhe, Germany, and received the PhD degree from Cornell University in 1997. He is an associate professor of computer science at Middlebury College, Vermont. His research interests include computer vision, particularly stereo vision, and robotics. He maintains several online stereo vision and optical flow benchmarks at http://vision.middlebury.edu. He is a member of the IEEE and the IEEE Computer Society.

Olga Veksler received the BS degree (with honors) in mathematics and computer science from New York University in 1995 and the PhD degree in computer science from Cornell University in 1999. From 1999 to 2003, she was a scientist at NEC Research Institute, Princeton, N.J. Since 2003, she has been an assistant professor of computer science at the University of Western Ontario, Canada. Her research interests lie in combinatorial optimization methods for computer vision, with applications to stereo correspondence, multiview reconstruction, and image segmentation. She is a member of the IEEE and the IEEE Computer Society.

Vladimir Kolmogorov received the MS degree in applied mathematics and physics from the Moscow Institute of Physics and Technology in 1999 and the PhD degree in computer science from Cornell University in 2003. After spending two years as an associate researcher at Microsoft Research, Cambridge, he joined University College London as a lecturer. He works on optimization algorithms for Markov random fields and their applications to stereo, image segmentation, and other vision problems. Two of his papers received a best paper award at the Seventh European Conference on Computer Vision (ECCV '02), and another paper received a best paper honorable mention award at the Conference on Computer Vision and Pattern Recognition. In 2006, he received the five-year Royal Academy of Engineering/EPSRC Research Fellowship. He is a member of the IEEE and the IEEE Computer Society.

Aseem Agarwala received the BS and MEng degrees at Massachusetts Institute of Technology in 1999 and the PhD degree from the University of Washington in 2006. He is a research scientist at Adobe Systems, Inc., and an affiliate assistant professor in the Computer Science and Engineering Department, University of Washington. His work experience includes two years as a research scientist at Starlab, Brussels, and research internships at MERL and Microsoft Research. His areas of research are computer graphics, computer vision, and computational imaging. He received the Honorable Mention (runner-up) for the 2006 ACM Doctoral Dissertation Award and a 2004 Microsoft Research Fellowship. His work can be found in several products, including Adobe Photoshop and Adobe Photoshop Elements. He is a member of the IEEE and the IEEE Computer Society.

Marshall Tappen received the PhD degree from the Massachusetts Institute of Technology. He is an assistant professor at the University of Central Florida, Orlando. His interests focus on the of intersection computer vision, image processing, and machine learning. He is a member of the IEEE and the IEEE Computer Society.

Carsten Rother received the diploma degree in computer science from the Universität Karlsruhe, Germany, in February 1999 and the PhD degree in computer vision from the Royal Institute of Technology (KTH), Stockholm, in January 2001. His thesis "Multi-View Reconstruction Using a Reference Plane" was nominated as one of the two Swedish candidates for the Best Nordic Thesis 2003-2004 Award. He also received a best paper honorable mention award at the 2005 IEEE CS Conference on Computer Vision and Pattern Recognition (CVPR). Since 2001, he has been with Microsoft Research, Cambridge. His main interests are in the area of image editing and synthesis, optimization algorithms for Markov random fields, 3D reconstruction, and object recognition. He is a member of the IEEE and the IEEE Computer Society.