University of Tokyo
Graduate School of Information Science and Technology
Department of Information and Communication Engineering

Applying Graph Cuts to MAP Estimation of Continuous and Higher-Order Markov Random Fields

written by

Tatsunori TANIAI

A M.Sc. Dissertation presented to the Graduate School of Information Science and Technology of University of Tokyo in partial fulfillment of the requirements for the degree of Master of Science in Information Science and Technology.

Advisor: Professor Takeshi NAEMURA

FEBRUARY 2014

Abstract

Markov random fields (MRFs) have been becoming the de facto standard probabilistic model for low-level vision tasks such as stereo vision and segmentation. In this approach the problems are formulated as the minimization of energy functions, which can be stochastically interpreted as a maximum-a-posteriori (MAP) estimation of MRF models. In principle, the use of more realistic MRF models is accompanied by more difficult energy optimization problems. Therefore, the performance of energy optimization methods has a significant impact on the overall performance of individual application methods.

In this dissertation we study the use of a discrete optimization method, graph cuts, for the energy minimization of the following two types of MRFs with appropriate applications: (i) pairwise MRFs with a continuous and multidimensional solution value space, which are effective for stereo vision; (ii) higher-order MRFs with a discrete solution value space, which are effective for segmentation.

In the first topic, we present an efficient inference method for pairwise MRFs in application to stereo matching problems, where a local 3D plane is estimated for each pixel in order to achieve accurate stereo vision. Despite the huge solution value space, the proposed method efficiently finds good approximate solutions. We appropriately design the proposed inference scheme so that it takes advantage of inherent properties of graph cuts and, at the same time, it accounts for the specifics of the considered stereo problems. Our method is evaluated on a standard stereo benchmark and achieves first place among more than 150 stereo algorithms.

In the second topic, we propose an efficient inference method for higher-order MRFs in application to image segmentation problems. We present the proposed method as an extension of two prior methods, by revealing their close connections. Our method can be used for various kinds of higher-order terms and achieves about an order of magnitude greater accuracy than the current state-of-the-art method. We further show that our method can be applied to multiple-labeling problems. To the best of our knowledge, this is the first work that shows viable solutions to multiple distribution matching problems. We also show that, in some reasonable settings, our method can yield good approximate solutions in only a single minimum cut operation, while it usually costs several minimum cut operations.

Keywords: energy minimization, Markov random field, higher-order energy, continuous stereo matching, segmentation

内容梗概

マルコフ確率場は，ステレオマッチングやセグメンテーションなどの低レベルのビジョン問題を扱う際の標準的な確率モデルになりつつある．このアプローチでは，問題がエネルギー関数最小化の枠組みで定式化され，これはマルコフ確率場上での最大事後確率推定（MAP 推定）として解釈される．このとき基本的に，より現実に即したモデルを用いるほど伴うエネルギー最適化問題は複雑になる．したがって，エネルギー最適化手法の性能は，それを用いる応用手法の最終的な性能を決定づける大きな要素になる．

本研究は，グラフカットと呼ばれる離散最適化手法を用いたエネルギー最適化手法について論じる．なかでも本研究で取り扱うのは，以下に述べる 2 種類のマルコフ確率場モデルについてであり，各モデルに対して適切な応用を設定して手法を提案する．(1) 連続値多次元の 1 階マルコフ確率場モデルに基づいたステレオマッチング問題．(2) 離散値多値の高階マルコフ確率場モデルに基づいた画像セグメンテーション問題．

最初の問題設定では，物体表面を近似する 3 次元平面を，各画素に対して密に推定するステレオモデルを扱う．3 次元平面を推定することで高精度なステレオマッチングが実現できるが，代わりに解の探索範囲が広大になる．提案する最適化手法は，このような場合でも効率的に良い近似解を求めることができる．提案手法は，グラフカット固有の性質を活かしつつ，同時にステレオ特有の性質に即するように適切に設計されたもので，150 以上のステレオ手法が登録された標準的ベンチマークにおいて 1 位の性能を達成した．

2 番目の問題設定では，画像セグメンテーションで用いられる色分布マッチングなどの様々な種類の高階関数を，効率的かつ汎用的に最適化する手法を提案する．ここではまず，2 つの先行研究の密接な関係が示され，提案手法はそれら 2 つの手法の一般化手法として提示される．提案手法は，最新の手法と比べて精度がおよそ 1 桁上回った．さらに提案手法を応用して，多値ラベリング問題に適用できることを示した．これは我々の知る限り，色分布マッチング手法を一般多値問題に適用した初めての例である．また，通常は良い解を得るのに何度もグラフカットを適用する必要があるが，提案手法を応用すると，ある条件設定において，たった 1 回の適用で良い近似解が得られることを示した．

Keywords: エネルギー最適化，マルコフ確率場，高階エネルギー，連続値ステレオマッチング，セグメンテーション

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

GC	graph cut
BP	belief propagation
PMBP	PatchMatch belief propagation
MRF	Markov random field
MAP	maximum a posteriori
QPBO	quadric pseudo-boolean optimization
CPU	central processing unit
GPU	graphics processing unit
AC	auxiliary cuts
SSP	submodular-supermodular procedure
SC	semidifferential cuts
BJ	Boykov and Jolly

1 Introduction

1.1 Markov Random Fields in Computer Vision

In computer vision, Markov random fields (MRFs) [Geman and Geman, 1984] have been becoming the de facto standard probabilistic models for low-level vision problems such as stereo vision and segmentation.

In this approach, solution values we seek are formulated as a mapping function $f_p = f(p) : \mathcal{P} \rightarrow \mathcal{S}$ that assigns each site $p \in \mathcal{P}$ some value from the solution value space $\mathcal{S}$. For example, in stereo vision, f represents a depth map on the image domain and we seek a depth value f_p for each pixel p from the depth value space $\mathcal{S}$. Likewise, in segmentation, f represents a segmentation labeling of the image domain and wee seek an object label f_p for each pixel p from the object label space $\mathcal{S}$. See Figures 1.1 and 1.2 for the visualization of f in stereo vision and segmentation.

The problems using MRFs are then formulated as the minimization of the following energy function:

$$E(f) = \underbrace{\sum_{p \in \mathcal{P}} \phi_p(f_p)}_{\text{unary terms}} + \underbrace{\sum_{(p,q) \in \mathcal{N}} \psi_{pq}(f_p, f_q)}_{\text{pairwise terms}} + \underbrace{\sum_{c \in \mathcal{C}} \phi_c(f_{c_1}, f_{c_2}, ..., f_{c_n})}_{\text{higher-order terms}} . \tag{1.1}$$

The first term is often called *data term*, because it is usually used for measuring sitewise consistencies between given data and the resulting solution value f_p of each site p. Likewise, the second term is often called *smoothness term*, because it is mostly used for enforcing smoothness on, *e.g.*, depth maps or segmentation labelings between neighboring pixels p and q. The third term describes simultaneous interactions of multiple sites and it is often used as a more advanced and realistic model for data terms [Rother *et al.*, 2006; Ayed *et al.*, 2013] and smoothness terms [Jegelka and Bilmes, 2011; Kohli *et al.*, 2013]. When $E(f)$ has only unary and pairwise terms, such formulations are called *pairwise MRFs*. Pairwise MRFs are widely used in many applications for their well-balanced trade-offs between complexity and description capability. When $E(f)$ contains higher-order terms, such formulations are called *higher-order MRFs*. The use of higher-order MRFs is usually expensive in terms of computational complexity, but it often

brings outstanding performances unattainable with standard pairwise MRF formulations.

Figure 1.1 Illustration of stereo vision, where a solution f is a depth map (right figure).

Figure 1.2 Illustration of image segmentation, where a solution f is an object labeling (right figure).

1.1.1 MAP Estimation and Energy Optimization using Graph Cuts

The minimization of such MRF functions is related to *maximum-a-posteriori* (MAP) estimation [Geman and Geman, 1984] and is thus called MAP-MRF estiation. The computational difficulty of MAP-MRF estimation depends on the forms of both the energy function $E(f)$ (*e.g.*, pairwise or higher-order) and the solution value space $\mathcal{S}$ (*e.g.*, discrete or continuous). In principle, more realistic models are accompanied by more difficult energy optimization problems. Therefore, the performances of energy optimization methods determine the maximum performance of *e.g.* stereo and segmentation methods.

Of various energy optimization approaches, we in this thesis focus on the study of the energy optimization methods using graph cuts (GC) [Boykov *et al.*, 2001; Kolmogorov and Zabin, 2004]. A notable ability of GC is that, the energy function $E(f)$ can be *exactly* and *globally* minimized in polynomial times via GC, if the solution value space is binary $\mathcal{S} = \{0, 1\}$ and $E(f)$ is expressed by a pairwise submodular form of MRFs [Kolmogorov and Zabin, 2004]. Taking advantage of this ability of GC, we propose two optimization methods using GC, for stereo vision and segmentation

respectively, that can efficiently find approximate yet good solutions for some difficult types of MRF energy functions.

1.2 Contributions

In this thesis, we study MAP estimation of the following two types of MRFs with appropriate applications: (i) pairwise MRFs $E(f)$ with a continuous and multidimensional solution value space $\mathcal{S} \subset \mathbb{R}^d$, which are effective for achieving accurate stereo vision; (ii) higher-order MRFs $E(f)$ with a discrete multiple-label space $\mathcal{S} = \{1, 2, ..., K\}$, which are effective for achieving accurate segmentation. The contributions of individual works are summarized in the following sections.

1.2.1 Efficient MAP Estimation of Continuous MRFs for Stereo Vision

We propose an efficient optimization method using GC for pairwise MRF based stereo vision. Here, a 3D label $f_p = (a_p, b_p, c_p)$ representing a local surface plane $d_p = a_p x + b_p y + c_p$ is estimated for each pixel p for achieving accurate stereo vision. The contributions of this work are summarized as follows:

- We propose an inference scheme that efficiently finds good approximate solutions from the huge solution value space.
- We appropriately design the proposed inference scheme so that it takes advantage of inherent properties of GC and at the same time accounts for the specifics of the considered stereo problems.
- The proposed method achieves first place among more than 150 stereo algorithms in a standard stereo benchmark [Scharstein and Szeliski, 2001].

1.2.2 Efficient MAP Estimation of Higher-Order MRFs for Segmentation

We propose an efficient optimization method using GC for higher-order MRF based segmentation. The proposed method is applicable to various kinds of useful higher-order terms including color-distribution matching terms such as L_p-distance, KL divergence, and Bhattacharyya measures. It can be also used for multiple-labeling problems as well as binary-labeling problems. The contributions of this work are summarized as follows:

- We point out a theoretically close relationship between two prior methods by Ayed *et al.* [2013] and Narasimhan and Bilmes [2005], and propose a generalized method by extending both approaches.
- The proposed method achieves about an order of magnitude greater accuracy than the current state-of-the-art method [Ayed *et al.*, 2013].

- We applied the proposed method to general multiple-labeling problems. To the best of our knowledge, this is the first work that shows viable solutions to multiple distribution matching problems.
- We show that, in some reasonable situations, our method can yield good approximate solutions to higher-order MRF energies in only a single GC operation, while it usually costs several minimum cut operations.

2
Background

2.1 Stochastic Relationship with MAP Estimation

The minimization of $E(f)$ in Equation (1.1) can be interpreted as MAP estimation [Geman and Geman, 1984]. In MAP estimation, stochastically plausible solutions are estimated by maximizing a posteriori probability $P(f|d) \propto P(d|f)P(f)$ with given observed data d. Below we briefly review the relationship between the minimization of $E(f)$ and MAP estimation using pairwise MRF formulations. To see this, we use the following equivalence:

$$f^* = \underset{f}{\operatorname{argmax}}\, P(f|d) \tag{2.1}$$

$$= \underset{f}{\operatorname{argmin}} - \log P(f|d) \tag{2.2}$$

$$= \underset{f}{\operatorname{argmin}} - \underbrace{\log P(d|f)}_{\text{likelihood}} - \underbrace{\log P(f)}_{\text{prior}}. \tag{2.3}$$

For the posteriori distribution $P(d|f)P(f)$, we use the Gibbs distribution forms with unary and pairwise potentials $\phi_p(f_p)$ and $\psi_{pq}(f_p, f_q)$:

$$P(d|f)P(f) \propto \left[\prod_{p\in\mathcal{P}} \exp\left(-\phi_p(f_p)\right)\right]\left[\prod_{p\in\mathcal{P}}\prod_{q\in\mathcal{N}(p)} \exp\left(-\psi_{pq}(f_p, f_q)\right)\right]. \tag{2.4}$$

By using this expression, we can derive the relationship between the posteriori probability $P(f|d)$ and energy function $E(f)$ as below:

$$\underset{f}{\operatorname{argmax}}\, P(f|d) = \underset{f}{\operatorname{argmin}}\, E(f). \tag{2.5}$$

In MAP-MRF estimation, the unary terms $\phi_p(f_p)$ and pairwise terms $\psi_{pq}(f_p, f_q)$ in $E(f)$ can be therefore interpreted as likelihood $-\log P(d|f)$ and prior $-\log P(f)$, respectively.

2.2 Discrete and Continuous Pairwise MRFs

There are two types of pairwise MRF formulations depending on whether the label space $\mathcal{S}$ is *discrete* or *continuous*.

When each node is assigned a discrete value, such models are often called discrete MRFs. There are many powerful discrete optimizers for minimizing such MRF energies. For example, message passing methods such as belief propagation (BP) [Yedidia *et al.*, 2000; Felzenszwalb and Huttenlocher, 2004] and tree re-weighted message passing [Kolmogorov, 2006], and combinational methods such as graph cuts (GC) [Boykov and Kolmogorov, 2004; Kolmogorov and Zabin, 2004] are of common choices. GC differs from message passing methods in that it improves all nodes simultaneously, and such a global property brings a good convergence ability in GC-based optimization by helping to avoid trapped at bad local minimas. Comparative studies on various discrete optimizers for various vision problems can be found in [Kappes *et al.*, 2013; Szeliski *et al.*, 2008], where GC with the expansion algorithm [Boykov *et al.*, 2001] has shown successful results. One of the biggest advantages of using discrete optimizers is that they can optimize non-convex energy functions, which are often useful in vision applications because they are very robust to outliers. On the other hand, the use of discrete label space is not appropriate in some applications such as stereo matching and optical flow, because *e.g.* in stereo the true scene depths reside in the continuous domains.

A simple solution to such continuous MRF estimation would be to use discrete optimizers with a finely discretized label space; however, this approach is computationally intractable due to the huge (infinite) label space. Practical choices are the use of continuous optimizers and discrete-continuous approaches. Continuous optimizers use various techniques developed in the context of convex optimization [Boyd and Vandenberghe, 2004; Nocedal and Wright, 2006] for efficiently and directly estimating continuous MRFs. Although they are very powerful tools for convex energy functions, the inference can be easily trapped at a bad local minima when optimizing highly non-convex functions [Lempitsky *et al.*, 2008]. Discrete-continuous approaches use discrete optimizers for estimating continuous MRFs in efficient manners. For example, fusion optimization [Lempitsky *et al.*, 2010] estimate continuous MRFs by combining a number of continuous-valued "weak solutions" using GC. PMBP [Besse *et al.*, 2012] incorporates the spatial propagation technique of PathMatch [Barnes *et al.*, 2009, 2010] for accelerating BP-based inference for continuous MRFs.

In Table 2.2, we summarize some basic optimization methods using GC for discrete and continuous pairwise MRFs. We describe each method in the following sections.

Table 2.1 Summary of GC-based optimization methods. The standard GC can be used for only submodular energies, whereas QPBO-GC can be used for submodular and non-submodular energies. For discrete multi-labeling problems the expansion move algorithm is useful. When the label space is huge (*e.g.*, label space is continuous), the fusion move algorithm is a practical choice.

Methods	Type of MRFs for main use	Output labels f_p
GC (min-cut)	binary submodular	$\{0, 1\}$
GC + expansion moves	discrete submodular	$\{0, 1, ..., N-1\}$
QPBO-GC	binary non-submodular	$\{0, 1, \emptyset\}$
QPBO-GC + expansion moves	discrete non-submodular	$\{0, 1, ..., N-1\}$
QPBO-GC + fusion moves	continuous non-submodular	$\{g_p^{(0)}, g_p^{(1)}, ..., g_p^{(N-1)}\}$

2.3 Inference Methods using GC for Discrete MRFs

2.3.1 S-T Minimum Cut for Binary MRFs

The essential function of GC is to minimize binary-labeling MRF energies, *i.e.*, $E(f)$ with the binary label space $\mathcal{S} = \{0, 1\}$. When all pairwise potentials in $E(f)$ meet the following condition known as *submodularity*

$$\psi(0,0) + \psi(1,1) \leq \psi(1,0) + \psi(0,1), \tag{2.6}$$

the minimization of $E(f)$ can be replaced with a min-cut / max-flow problem in the graph theory, which can be *optimally* and *exactly* solved in a polynomial time [Kolmogorov and Zabin, 2004]. Specifically, the s-t minimum cut is performed on a graph illustrated in Figure 2.1, resulting in a s-t partition (S, T) where $S \subseteq \mathcal{P}$ and $T = \mathcal{P} \setminus S$. The globally optimal labeling $f^* = \text{argmin}\, E(f)$ is then obtained by the following rules:

$$f_p \leftarrow \begin{cases} 1 & \text{if } p \in S \\ 0 & \text{if } p \in T \end{cases}. \tag{2.7}$$

This binary formulation is directly used in foreground-background image segmentation [Boykov *et al.*, 2001; Rother *et al.*, 2004].

2.3.2 Expansion Moves for Discrete MRFs

Boykov *et al.* [2001] proposes the expansion move algorithm for efficiently optimizing multi-labeling MRFs using GC. Algorithm 2.1 shows the overview of the expansion move algorithm. In this method the multi-labeling problem is reduced to a sequence of binary-labeling problems, where each node p is assigned either its current label f_p or a proposal label $\alpha \in \mathcal{S}$. This sequential process is also illustrated in Figure 2.2. The line 5 of Algorithm 2.1 shows the binary problems, where the binary-labeling MRF energies are minimized via the s-t minimum cut by applying the

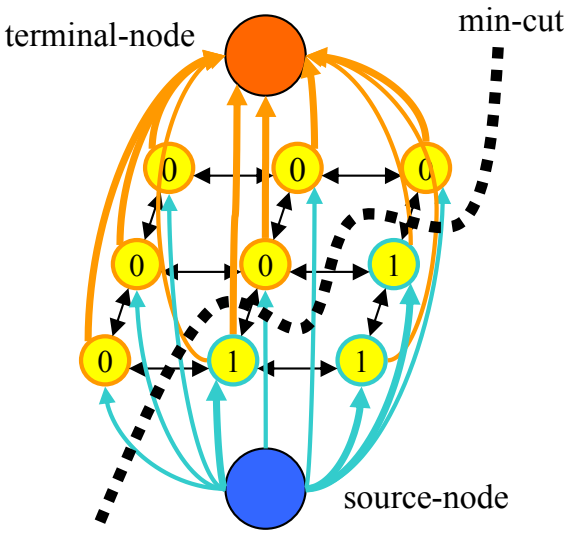


Figure 2.1 Illustration of s-t minimum cut for 2D grids. In GC optimization, we make a graph with pixel nodes and two spatial nodes called source and terminal nodes. We encode the unary costs into the edge weights between each pixel node and the source/terminal node (blue/orange edges), and the pairwise costs into the edge weights between neighboring pixel nodes (black edges). The optimal labeling is obtained as the minimum-cost cut that separates pixel nodes into either source or terminal side.

Algorithm 2.1 Expansion algorithm

1: Define the discrete label space $\mathcal{S} := \{s^{(0)}, s^{(1)}, ..., s^{(N-1)}\}$
2: Initialize f by setting $f_p \leftarrow s^{(0)}$
3: **repeat**
4: **for all** labels $\alpha \in \mathcal{S}$ **do**
5: $f \leftarrow \text{argmin}\, E(f' | f'_p \in \{f_p, \alpha\})$
6: **end for**
7: **until** convergence

following update-rules:

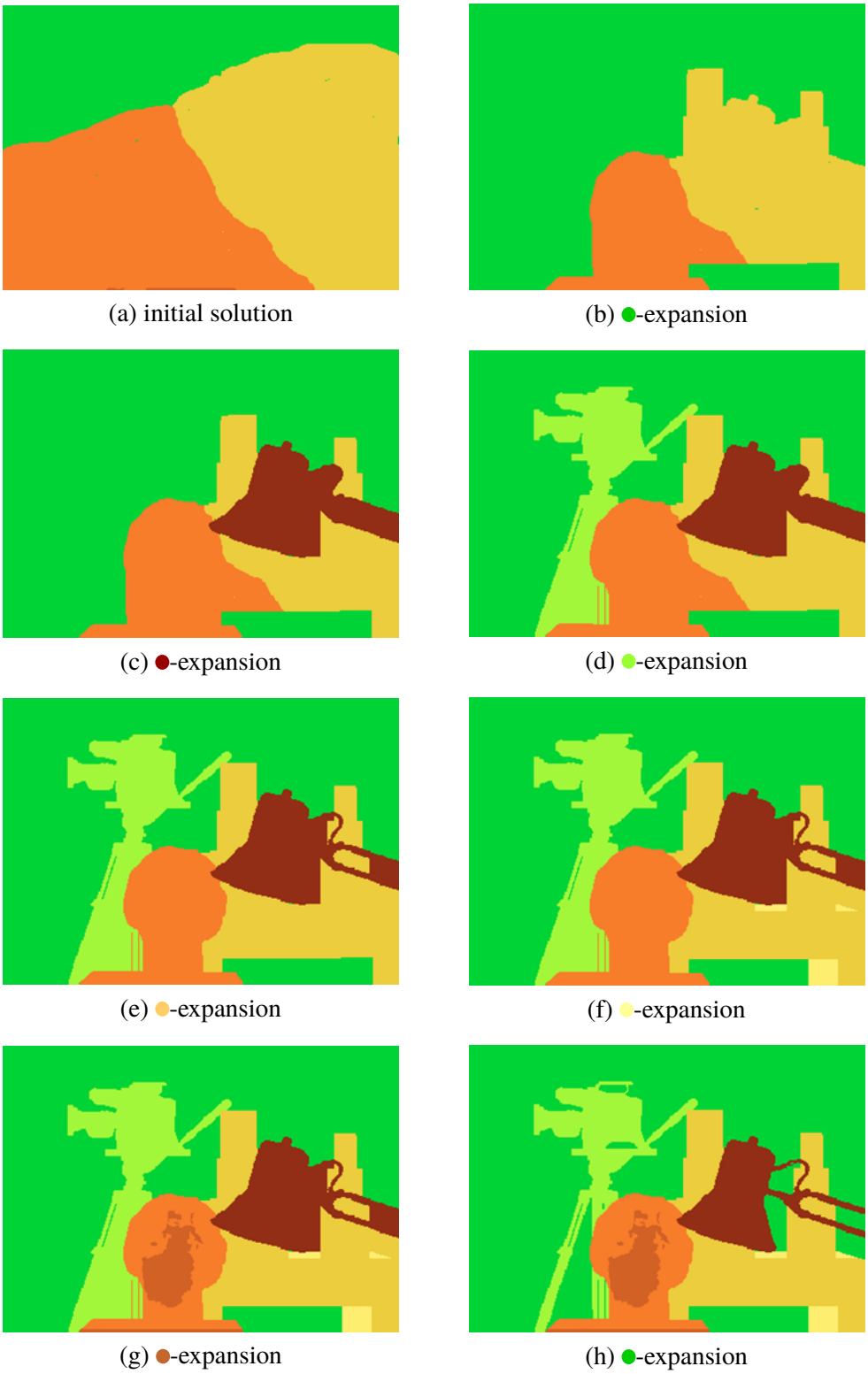
$$f_p \leftarrow \begin{cases} \alpha & \text{if } p \in S \\ f_p & \text{if } p \in T \end{cases} \tag{2.8}$$

Here, each binary problem is *optimally* solved by the s-t minimum cut, if only pairwise potentials ψ meet the following submodularity of expansion moves [Boykov *et al.*, 2001; Kolmogorov and Rother, 2007]:

$$\psi(\alpha, \alpha) + \psi(\beta, \gamma) \leq \psi(\beta, \alpha) + \psi(\alpha, \gamma). \tag{2.9}$$

2.3.3 QPBO for Non-Submodular Binary MRFs

So far, we have only discussed the cases where the energy functions and move-energies are submodular. Actually, we cannot correctly treat non-submodular energy functions with standard GC



(a) initial solution (b) ●-expansion

(c) ●-expansion (d) ●-expansion

(e) ●-expansion (f) ●-expansion

(g) ●-expansion (h) ●-expansion

Figure 2.2 Illustration of the expansion move algorithm [Boykov *et al.*, 2001]. Starting with (a) an initial solution, the expansion algorithm successively expands the region of a proposal label as shown in (b) to (h). In other words, in each expansion operation, each pixel's solution is allowed to move to the proposal label or stay at its current label. The images are cited from [Boykov *et al.*, 2007].

optimization because they are not "graph representable" [Kolmogorov and Zabin, 2004] [1]. Recently, Kolmogorov and Zabih [2001] have introduced *quadric pseudo-boolean optimization* (QPBO) or so-called QPBO-GC [Boros *et al.*, 1991; Hammer *et al.*, 1984] into the vision community, which has become a standard technique for handling non-submodular energies in GC optimization.

QPBO-GC is essentially an optimization technique for binary MRFs. In QPBO-GC, each node is duplicated as p and its inverse node $\bar{p}$, and the s-t minimum cut is performed on a special graph using the duplicated nodes. Its output is given as a *partial labeling* $f_p \in \{0, 1, \emptyset\}$ by the following rules:

$$f_p \leftarrow \begin{cases} 1 & \text{if } p \in S \text{ and } \bar{p} \in T \\ 0 & \text{if } p \in T \text{ and } \bar{p} \in S \\ \emptyset & \text{otherwise} \end{cases}, \tag{2.10}$$

where the label $\emptyset$ means that the nodes are left *unlabeld*. It is guaranteed that if f_p is labeled either 0 or 1, the optimal solution is partially given as $f_p^* = f_p$ for the labeled nodes p. Furthermore, it is guaranteed that the optimal solution f^* is obtained when the energy functions are submodular.

2.4 Inference Methods for Continuous MRFs

2.4.1 Fusion Moves for Continuous MRFs

The introduction of QPBO-GC has brought a powerful GC-based optimization scheme, fusion moves [Lempitsky *et al.*, 2010], for continuous MRFs. Fusion moves are an operation that combines two solutions f and g to make a better solution, where each node p is assigned either f_p or g_p. Here, "better" means that the energy value of the fusion result is not higher than both $E(f)$ and $E(g)$. We call this operation "binary fusion". Note that expansion moves [Boykov *et al.*, 2001] are special cases of fusion moves where g is given spatially constant as $g_p = \alpha$. The fusion-based optimization therefore proceeds similarly to the expansion move algorithm as shown in Algorithm 2.2.

Algorithm 2.2 Fusion algorithm

1: Generate a set of solution proposals $G := \{g^{(0)}, g^{(1)}, ..., g^{(N-1)}\}$
2: Initialize f by setting $f_p \leftarrow g_p^{(0)}$
3: **repeat**
4: **for all** proposals $g \in G$ **do**
5: $f \leftarrow \text{argmin } E(f' | f'_p \in \{f_p, g_p\})$
6: **end for**
7: **until** convergence

In the fusion-based optimization frameworks, a number of plausible solutions $\{g^{(0)}, g^{(1)}, ..., g^{(N-1)}\}$

[1] When binary-labeling MRF energies are non-submodular, the corresponding s-t graphs contain negative edge weights, for which the min-cut / max-flow algorithm cannot be applied.

or so-called *proposals* are first generated by other external methods. The proposals are then successively combined as a sequence of binary fusions, where each binary fusion combines the current solution f and one of proposal solutions g. Since the binary fusions can contain non-submodular terms, the minimization at the line 5 of Algorithm 2.2 is performed via QPBO-GC by the following rules:

$$f_p \leftarrow \begin{cases} g_p & \text{if } p \in S \text{ and } \bar{p} \in T \\ f_p & \text{if } p \in T \text{ and } \bar{p} \in S \\ f_p & \text{otherwise} \end{cases}. \tag{2.11}$$

In the third case of the above equation, unlabeled nodes are set to their current labels f_p, by doing which the energy value $E(f)$ is guaranteed not to increase throughout the iterations [Kolmogorov and Rother, 2007]. The final fusion result f is assigned as $f_p \in \{g_p^{(0)}, g_p^{(1)}, ..., g_p^{(N-1)}\}$ so as to minimize $E(f)$. Therefore, we can use the fusion technique for optimizing continuous MRFs by using continuous-valued proposals. The spirit of the fusion approaches here is to use many "weak" solutions to optimize difficult and complex models.

The fusion algorithm in Algorithm 2.2 is sometimes described as "fusion-based expansion algorithm" or "fusion using the expansion algorithm". This is because the fusion of multiple proposals can be regarded as a multi-labeling problem of assigning a "meta-label" $i_p \in \{0, 1, .., N-1\}$ for each pixel p such that the fusion solution f given by $f_p = g_p^{(i_p)}$ minimizes $E(f)$; and in this view, we can say that Algorithm 2.2 solves the multi-labeling problem using the expansion algorithm on the meta-labels $\{0, 1, .., N-1\}$. To avoid the confusion, however, we strictly distinguish expansion moves from fusion moves; we refer to fusion moves as expansion moves only when proposals $\{g^{(i)}\}$ are given spatially constant, *i.e.*, $g_p^{(i)} = \alpha$ holds for all pixels $p \in \mathcal{P}$. This difference derives a subproblem optimality for expansion moves such that each binary-labeling problem during the optimization (*i.e.* the line 5 of Algorithm 2.1) can be optimally solved without employing expensive QPBO-GC if only the submodularity of Equation (2.9) is satisfied, which in contrast is not the case for general fusion moves.

Although the fusion algorithm of Algorithm 2.2 is a standard fashion for fusing multiple proposals, fusion itself can be achieved using other than the sequential fashion of the expansion algorithm. A possible choice is LogCut [Lempitsky *et al.*, 2007], where the proposals are fused not sequentially but hierarchically just like a tournament tree. In LogCut the number of graph cuts required for vising all proposals grows only logarithmically w.r.t. the number of the proposals, whereas the complexity is linear in the expansion algorithm.

2.4.2 PAERL Algorithm for Multi-Model Fitting

Multi-model fitting [Isack and Boykov, 2012; Delong *et al.*, 2012] is a problem of segmenting data points with geometric models, *e.g.* fitting planes to point cloud data, and we consider that it takes a middle position between discrete and continuous MRF problems; in multi-model fitting, the label space we must search is continuous and multi-dimensional but the number of actually used

labels is small. For such problems, the PAERL algorithm [Isack and Boykov, 2012] presented in Algorithm 2.3 has shown to be useful.

Algorithm 2.3 PEARL algorithm

1: Initialize geometric models randomly $\mathcal{S} \leftarrow \{s^{(0)}, s^{(1)}, ..., s^{(N-1)}\}$
2: **repeat**
3: Assign geometric models: $f \leftarrow \text{argmin}\, E(f|\mathcal{S})$
4: Update geometric models: $\mathcal{S} \leftarrow \text{argmin}\, E(\mathcal{S}|f)$
5: **until** convergence

The PAERL algorithm searches the continuous label space by alternating between assigning and updating geometric models. The assignment of given geometric models is achieved by the expansion algorithm at line 3 of Algorithm 2.3. Then, at line 4, we update the geometric models by minimizing the energies with the labeling fixed at f; *e.g.* when fitting planes to point clouds, each plane model $s^{(i)} \in \mathcal{S}$ can be updated by least squares regression using inlier points that are assigned $s^{(i)}$.

Because we must visit all labels in the expansion algorithm, this approach is only viable when the number of geometric models in the scenes is small [2], and it becomes intractable with *e.g.* dense stereo matching that estimates per-pixel independent planes. In fact, Olsson and Boykov [2012] report that estimating dense planes for 10K points takes about three hours using a very similar optimization approach.

2.4.3 PatchMatch for Nearest-Neighbor Field Estimation

Nearest-neighbor field estimation is a kind of dense correspondence search between two images, but it considers no spatial smoothness but only matching similarity of patches; thus, it can be formulated as MRFs with only unary terms measuring patch-similarity.

PatchMatch [Barnes *et al.*, 2009, 2010] is an efficient inference method for nearest-neighbor field estimation using spatial propagation and randomized search. Its algorithm is summarized in Algorithm 2.4.

The idea behind spatial propagation is that if a well-matched correspondence is found at a pixel p, it has a very high chance that the p's solution gives good estimates for p's nearby pixels too. Using this idea, the PatchMatch algorithm first assigns random values to all pixels, and visits the pixels sequentially in raster-scan order. Then, at each pixel p, (1) it collects current labels assigned to p's neighboring pixels, (2) chooses the best label among p's current label and the neighbors' labels, (3) refines the chosen label using iterative randomized search and takes it as p's label. The search scope in this step (3) is reduced exponentially as shown in line 6 of Algorithm 2.4.

Despite its simpleness, PatchMatch works very successfully even for large or high-dimensional

[2] In fact, the number of actually used labels in the solutions f is directly reduced using so-called *label costs* in multi-model fitting [Delong *et al.*, 2012; Isack and Boykov, 2012].

Algorithm 2.4 PatchMatch algorithm

Initialize f randomly
repeat
  for all pixels $p \in \mathcal{P}$ in raster-scan order **do**
    $f_p \leftarrow \text{argmin}\, \phi_p(s)$ with $s \in \{f_p, f_q | q \in \mathcal{N}(p)\}$
    for $m = 1$ to M **do**
      $r \leftarrow f_p + \text{random} \in [0, \sigma^2/2^m]$
      $f_p \leftarrow \text{argmin}\, \phi_p(s)$ with $s \in \{f_p, r\}$
    end for
  end for
until convergence

label spaces. Although PatchMatch only optimizes unary terms, Besse *et al.* [2012] propose a unified method of PatchMatch and BP for pairwise MRFs with smoothness regularization terms.

3

MAP Estimation of Continuous MRFs for Stereo Vision

3.1 Introduction

3.1.1 Stereo Vision

Stereo vision is a technique for estimating the 3D geometry of static scenes using multiple images taken from different view points. Although the recent availability of consumer depth cameras has been gaining attention in computer vision, the image-based modeling approaches still have advantages because RGB-image cameras are much more low-cost and popular. For example, Agarwal *et al.* [2011] build a system that reconstructs the 3D geometries of a whole city using more than 100K images available on Internet photo-sharing sites. Furthermore, point clouds obtained by depth sensors are often very sparse due to the limited resolutions of depth cameras. Wang and Yang [2011] show that such sparse depth points can be refined and up-sampled by using the stereo vision techniques.

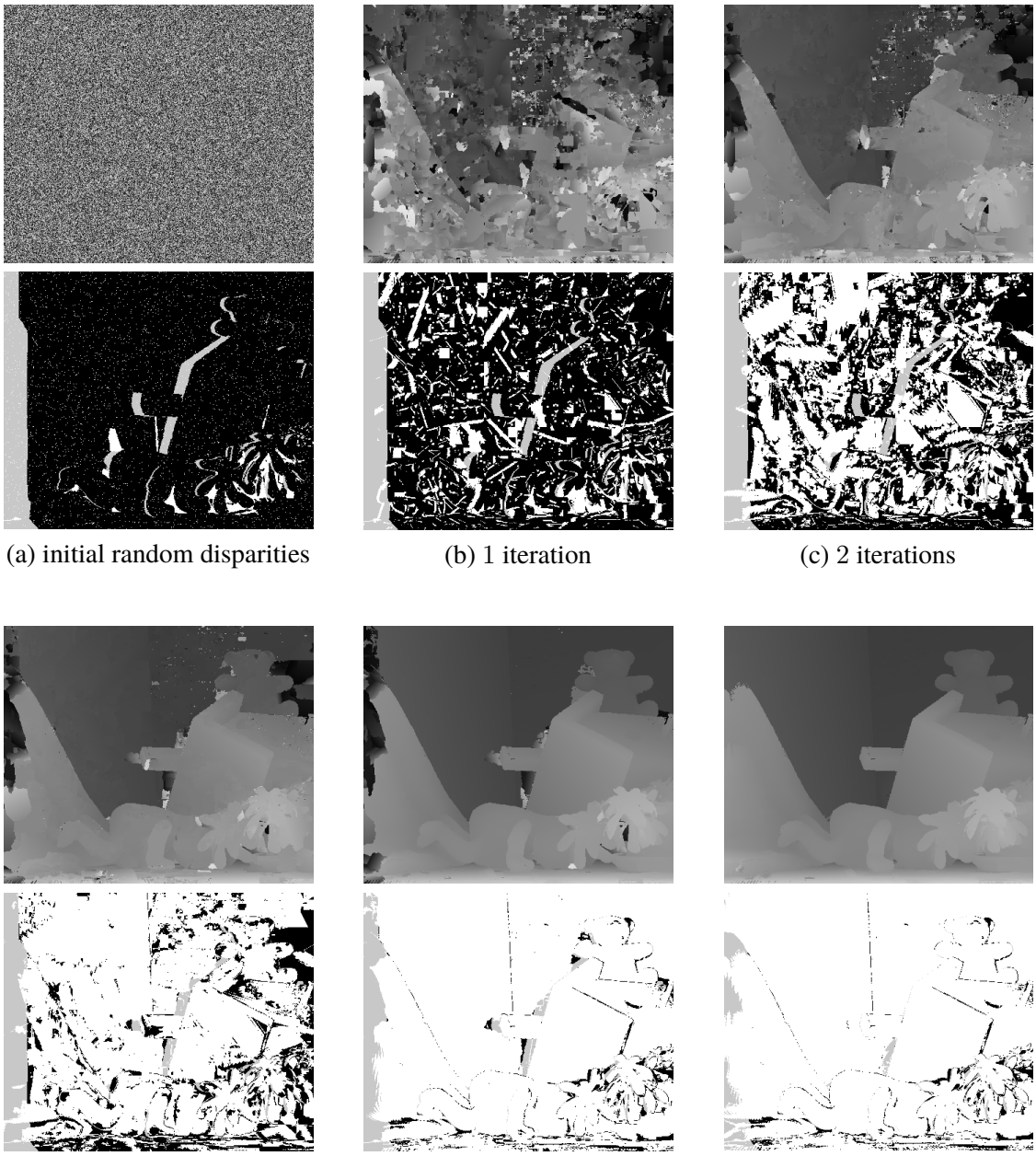
Of various settings of stereo vision, the binocular stereo problems, which assume two input images taken from horizontally-positioned parallel cameras, make a fundamental building block in stereo vision. Figure 3.2 depicts the typical settings of binocular stereo vision. Here, a depth value z at an image coordinate p on the left view image can be triangulated by

$$z = fB/d \tag{3.1}$$

if p's corresponding point $p' = p - (d, 0)^T$ on the right image as well as the camera's focal length f and baseline length B are known. Therefore, the objective of binocular stereo vision amounts to finding the best matching d or so-called *disparity* for each pixel.

3.1.2 Overview and Subjects of Stereo Vision

Stereo vision consists of several fundamental subjects, each of which is often studied as an independent topic. As an area overview we introduce representative subjects in stereo vision



(a) initial random disparities (b) 1 iteration (c) 2 iterations

(e) 3 iterations (f) 10 iterations (g) post-processing

Figure 3.1 Process of our stereo matching, showing disparity and error maps for Teddy. In our framework, we start with (a) random disparities that are represented by per-pixel 3D planes. We then alternately propagate local plane candidates and refine them in an iterative manner. (b)–(f) show the results after each propagation stage. Unlike previous methods, we use graph cuts for the propagations in an energy minimization framework. Finally, the resulting disparity map is further refined at (g) post-processing stage based on left-right consistency check and weighted median filtering.

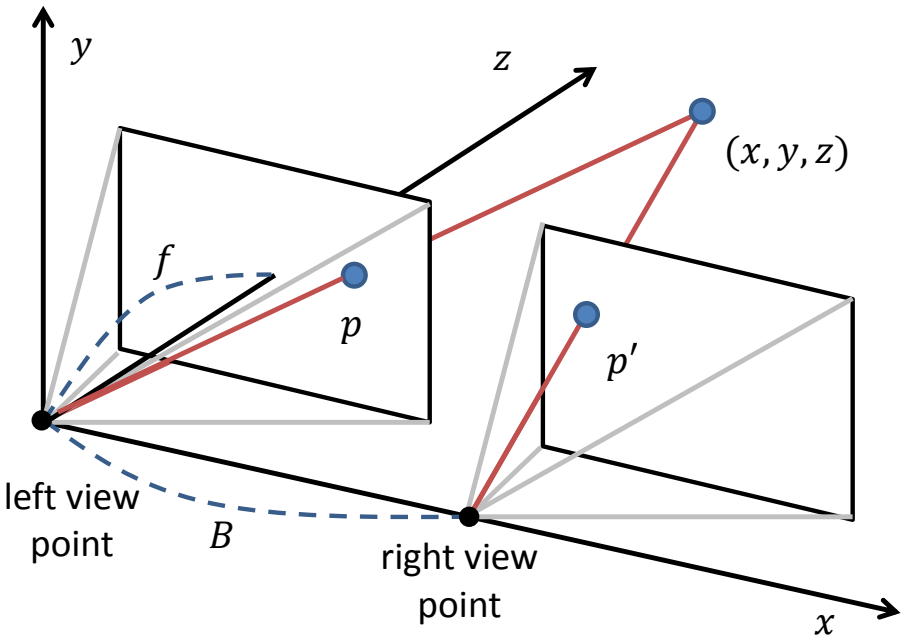


Figure 3.2 Basic setups of rectified binocular stereo vision. Two parallel cameras with the focal length f are placed at $(0, 0, 0)$ and $(B, 0, 0)$. A 3D point (x, y, z) is projected onto the left and right image planes at pixel coordinates p and p', respectively.

including optimization as our main focus in this paper. Please refer to [Hartley and Zisserman, 2004] for more details of the theory of multiview stereo.

Camera Calibration

Camera calibration is the estimation of *extrinsic* and *intrinsic parameters* of cameras. Extrinsic parameters are a translation T and rotation R of each camera, which denote the coordinate system transformations from 3D world coordinates to 3D camera coordinates. They can be estimated from images using structure-from-motion techniques [Hartley and Zisserman, 2004]. Intrinsic parameters such as focal length f, principal point, and lens distortion define the perspective projection transformations from 3D camera coordinates to image coordinates.

In binocular stereo vision, identical rotations and intrinsic parameters are assumed between a pair of cameras. Also, camera calibration is often omitted because as long as the performances are evaluated by disparity instead of depth, such calibration parameters are not required. In fact, a well-known online stereo benchmark known as the Middlebury stereo benchmark [Scharstein and Szeliski, 2001] evaluates accuracies by disparity. In this paper we omit this stage, or assume pre-calibrated cameras.

Rectification

Image rectification is a pre-processing technique for stereo vision. Given a calibrated camera pair, image rectification [Monasse, 2011] transforms input image pairs so that their non-parallel camera rotations are set parallel to each other as shown in Figure 3.2. By rectification, stereo matching problems are simplified because p and its corresponding point p' in Figure 3.2 have the same y-coordinate, and we only need to estimate horizontal disparity. We assume a rectified stereo image pair as input.

Photo-Consistency

Because stereo vision is a problem of finding correspondences between image pairs, the design of accurate and robust photo-consistency measures is a fundamental factor in stereo vision. Basically, we evaluate the reliability of matching points by measuring pixel intensity differences; for example, Birchfield and Tomasi [1998] propose pixel dissimilarity measure that is insensitive to image sampling. However, intensity-based measures are not robust to illumination changes. Therefore, gradient-based photo-consistency measures are often used by combining with intensity-based measures linearly [Klaus *et al.*, 2006] or selectively [Xu *et al.*, 2012].

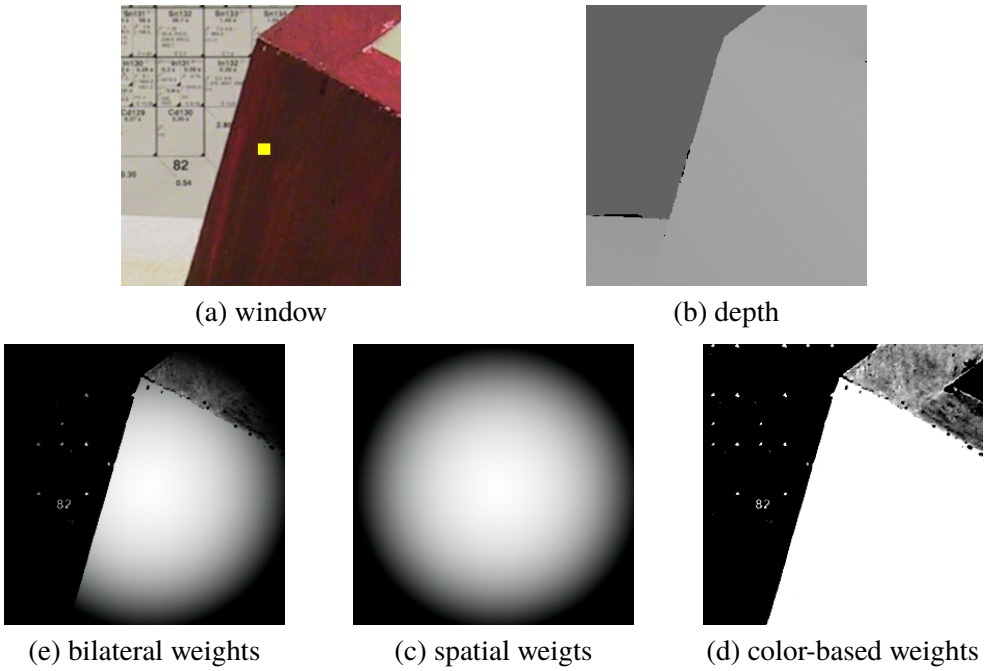
Cost Aggregation

If photo-consistency is measured using only single pixels, it will be fairly unreliable due to matching ambiguity and noises. Therefore, matching costs are usually aggregated, *i.e.*, multiple matching costs for nearby pixels are integrated to calculate each pixel's matching reliability. In other words, we use local windows for matching pixels.

There is an implicit assumption in this approach, *i.e.*, all pixels in a local window have the same disparity with that of the center pixel. This assumption is likely to collapse in two cases: (1) when pixels in the window lie on a different surface than the center pixel; (2) when the local region is not front-parallel and highly slanted. These are quite problematic because when we use larger windows for increasing the matching reliability, the constant-disparity assumption is more likely to collapse.

The first case is well handled by incorporating adaptive window approaches [Hosni *et al.*, 2012], where we adaptively assign weights for the pixels in the window based on the color similarity and spatial distance from the center pixel of the window. The concept is illustrated in Figure 3.3. To achieve edge-aware adaptive windows, edge-preserving filtering such as bilateral filtering [Tomasi and Manduchi, 1998; Yoon and Kweon, 2005] and guided image filtering [He *et al.*, 2013; Rhemann *et al.*, 2011] are used as cost-volume filtering. Therefore, the development of fast and highly edge-preserving filtering can be an important topic in stereo vision.

For the second case, Bleyer *et al.* [2011] recently show that, by assigning a local 3D disparity plane for each pixel, accurate photo-consistencies are measured avoiding the front-parallel biases. In this approach, however, the label space we must search for each pixel is too huge to be



(a) window (b) depth

(e) bilateral weights (c) spatial weigts (d) color-based weights

Figure 3.3 Concept of adaptive support windows. (a) The support window may contain pixels that lie on a different surface than the center (yellow) pixel, just as shown in (b) the depth map. The bilateral adaptive windows [Yoon and Kweon, 2005] define matching weights shown in (e) by combining (c) spatial weights and (d) color-based weights. In the context of cost-volume filtering the weights can be regarded as filter kernels.

tractable with the conventional exhaustive strategies. This problem imposes another difficultly in optimization, which we mainly focus on in this paper.

Regularization

When photo-consistency measures are ambiguous and noisy, the use of regularization can be very helpful. The smoothness regularization is usually used in stereo matching, which assumes and enforces spatially smooth disparity maps. The regularization is formulated as pixel interaction terms, which have to be optimized in a global manner. The simplest regularizer is picewise-constant models $|d_p - d_q|$ that penalize disparity deviations between neighboring pixels. However, because this model has a front-parallel bias and is not realistic to use, the second-order smoothness regularization is preferred in practice. Woodford *et al.* [2009] propose a second-order smoothness term in a form of $|d_r - 2d_p + d_q|$; however, as it involves three pixels into interactions, its optimization is rather complicated. Olsson *et al.* [2013] propose a second-order smoothness term based on pixelwise plane formulations, which can be achieved as pairwise interactions. We use their smoothness term in our proposed method.

Optimization

If stereo matching formulations contain pairwise or higher interaction terms, those cost functions must be optimized by a global optimization method such as belief propagation and graph cuts. Furthermore, if the disparity label space is high-dimensional or continuous, *e.g.*, disparity planes have a three-dimensional continuous label space, the optimization becomes even difficult because we cannot exhaustively search the entire label space with discrete optimizers. In this paper, we focus on the optimization of continuous pixelwise plane models.

Post-Processing and Refinement

The post-processing is used to refine resulting disparity maps by *e.g.* removing unreliable estimates by checking the consistency between left and right disparity maps, and applying edge-aware filtering. This process is often used for handling occlusions as described below.

Occlusions

Occlusions are problems particular to stereo vision. If a pixel in one image is invisible in the other images, the pixel is occluded and its disparity thus cannot be obtained in principle. In practice, we can estimate the depth of occluded pixels using the assumption that nearby pixels with similar colors tend to have a close depth value. Occlusions are handled mainly by two ways: during post-processing [Rhemann *et al.*, 2011], or optimization [Kolmogorov and Zabih, 2001, 2002; Wei and Quan, 2005; Woodford *et al.*, 2007].

3.1.3 Motivations and Contributions

Recent years have seen significant progress in accuracy of stereo vision. One of the breakthroughs is the use of 3D labels [Bleyer *et al.*, 2011; Besse *et al.*, 2012; Lu *et al.*, 2013; Olsson *et al.*, 2013; Heise *et al.*, 2013]; by estimating a local 3D disparity plane $d = a_p x + b_p y + c_p$ for each pixel, accurate photo-consistency is measured between matching pixels even with large matching windows. While stereo with standard 1D discrete disparity labels [Wang and Yang, 2011; Kolmogorov and Zabih, 2002, 2001; Boykov *et al.*, 2001] can be directly solved by discrete optimizers such as graph cuts (GC) [Kolmogorov and Zabin, 2004; Boykov and Kolmogorov, 2004] and belief propagation (BP) [Yedidia *et al.*, 2000; Felzenszwalb and Huttenlocher, 2004], such approaches cannot be directly used for continuous 3D labels due to the huge (infinite) label space $(a, b, c) \in \mathbb{R}^3$.

Recent successful methods [Bleyer *et al.*, 2011; Besse *et al.*, 2012; Lu *et al.*, 2013] use PatchMatch [Barnes *et al.*, 2009, 2010] to efficiently infer correct 3D planes using spatial propagation; each pixel's candidate plane is, in raster-scan order, refined and then propagated to next pixels. Further in [Besse *et al.*, 2012], this sequential algorithm is combined with BP yielding an efficient optimizer PMBP for pairwise Markov random fields (MRFs) [Geman and Geman, 1984]. In terms of MRF optimization, however, BP is considered a *sequential optimizer*, which improves each

node individually keeping others conditioned at the current state. In contrast, GC improves all nodes simultaneously by accounting for interactions across nodes, and this global property helps optimization avoid local minima [Szeliski *et al.*, 2008; Woodford *et al.*, 2009]. Nevertheless, incorporating spatial propagation into GC-based optimization is not straightforward, because inference using GC proceeds rather *all-nodes-simultaneously*, not *one-by-one-sequentially* like PatchMatch and BP.

In this paper, we introduce a new labeling scheme, *locally shared labels*, that enables spatial propagation in fusion-based optimization using GC [Lempitsky *et al.*, 2010]. The locally shared labels define, for each pixel or region, its compact and local discrete label space that is shared among neighboring pixels/regions. By using locally shared labels we generate a number of disparity maps or so-called *proposals* in the literature [Lempitsky *et al.*, 2010], and fuse and refine them in an iterative manner (see Figure 3.1). For natural scenes that often exhibit locally planar structures [1], the joint use of locally shared labels and GC has a useful property; it allows multiple pixels in a local region to be assigned the same disparity plane *by a single min-cut* in order to find smooth solutions and to avoid trapped at a bad local minima.

Advantages

The advantages of our method are fourfold:

- First, our locally shared labels produce *submodular moves* that guarantee the optimal labeling at each min-cut (subproblem optimal), which in contrast is not guaranteed in general fusion moves [Lempitsky *et al.*, 2010].
- Second, this optimality property and spatial propagation allow randomized search, rather than employ external methods to generate plausible initial proposals as done in previous fusion approaches [Lempitsky *et al.*, 2010; Woodford *et al.*, 2009; Olsson *et al.*, 2013], which may limit the possible solutions.
- Third, our method achieves greater accuracy than BP [Besse *et al.*, 2012] thanks to the good properties of GC and locally shared labels.
- Finally, unlike PMBP [Besse *et al.*, 2012] the computation of both unary and pairwise costs can be performed in a parallel manner[2], which is the most expensive part in practice. With the proposed approach, accurate stereo matching can be efficiently computed with a GPU implementation as we will see in the experiment.

[1] It has been shown in [Heise *et al.*, 2013] that a plane in the 3D world coordinates is still expressed as a plane in the disparity space on the 2D image domains.

[2] Although BP is usually GPU-parallelizable, PMBP differs from BP's standard settings in that it defines label space *uniquely and distinctively* for each pixel and *propagate* it; both make parallelization indeed non-trivial.

Table 3.1 Summary of MRF stereo methods. There is a trade-off relationship between the reconstruction quality and computational complexity in the three approaches of discrete, segment-based, and continuous stereo. Our work focuses on the optimization for the continuous stereo models.

Method types	Advantages	Disadvantages
discrete stereo	can use discrete optimizers	quantized depth
segment-based stereo	no depth quantization error	hard-segment, piecewise planar
continuous stereo	smooth surface	difficult to estimate

3.2 Related Work

MRF stereo methods can be categorized into three approaches: discrete stereo, segment-based stereo, and continuous stereo. A summary is given in Table 3.2.

Discrete Stereo

Discrete stereo [Wang and Yang, 2011; Kolmogorov and Zabih, 2002, 2001; Boykov *et al.*, 2001] formulates stereo matching as a discrete multi-labeling problem, where each pixel is individually assigned one of pre-defined discrete disparity values. For this problem, many powerful discrete optimizers, such as BP [Yedidia *et al.*, 2000; Felzenszwalb and Huttenlocher, 2004], tree re-weighted message passing [Kolmogorov, 2006], and GC [Kolmogorov and Zabin, 2004; Boykov and Kolmogorov, 2004], can be directly used. Successful results are shown using GC with expansion moves [Boykov *et al.*, 2001; Szeliski *et al.*, 2008].

Segment-based Stereo

Segment-based stereo [Tao *et al.*, 2001; Hong and Chen, 2004; Klaus *et al.*, 2006; Wang and Zheng, 2008] assigns a 3D disparity plane for each of over-segmented image regions. The candidate planes are generated by fitting planes to a roughly estimated disparity map, and then the optimal assignment of the planes is estimated by, *e.g.*, GC with expansion moves [Boykov *et al.*, 2001; Hong and Chen, 2004] or BP [Felzenszwalb and Huttenlocher, 2004; Klaus *et al.*, 2006]. Although this approach yields continuous-valued disparities, it strictly limits the reconstruction to a piecewise planar representation. Also, results are subject to the quality of the segmentation.

Continuous Stereo

The last group, to which our method belongs, is continuous stereo [Woodford *et al.*, 2009; Bleyer *et al.*, 2011; Besse *et al.*, 2012; Olsson *et al.*, 2013; Lu *et al.*, 2013; Heise *et al.*, 2013], where each pixel is assigned a distinct continuous disparity value. Some methods [Woodford *et al.*, 2009; Olsson *et al.*, 2013] use fusion moves [Lempitsky *et al.*, 2010], an operation that combines two disparity maps to make a better one (binary fusion) by solving a non-submodular binary-labeling problem using QPBO-GC [Kolmogorov and Rother, 2007; Lempitsky *et al.*, 2010]. In this

approach, a number of continuous-valued disparity maps (or proposals) are first generated by other external methods (*e.g.*, segment-based stereo [Woodford *et al.*, 2009]), which are then combined as a sequence of binary fusions. Our method is also based on fusion moves but generates proposals using locally shared labels, which enable spatial propagations of local candidate planes and, more importantly, they make fusion moves submodular, *i.e.*, each binary fusion is optimally solved via GC (subproblem optimal). Our method only requires randomized initial proposals instead of those generated by external methods.

A stereo method by Bleyer *et al*. [Bleyer *et al.*, 2011] proposes accurate photo-consistency measures using 3D disparity planes that are inferred by PatchMatch [Barnes *et al.*, 2009, 2010]. Heise *et al*. [Heise *et al.*, 2013] incorporates Huber regularization into [Bleyer *et al.*, 2011] using a joint framework of PatchMatch and convex optimization. Besse *et al*. [Besse *et al.*, 2012] point out a close relationship between PatchMatch and BP and present a unified method called PatchMatch BP (PMBP) for pairwise continuous MRFs. PMBP is probably the closest approach to ours in spirit, but we use GC instead of BP for the inference. Therefore, our method is able to take advantage of better convergence of GC [Szeliski *et al.*, 2008] for achieving greater accuracy. In addition, our method allows parallel computation of both unary and pairwise costs.

3.3 Proposed Method

This chapter describes the proposed stereo matching method. Given two input images I_L and I_R, our purpose is to estimate disparity maps of both images.

3.3.1 Formulation

We use a pairwise MRF formulation by following conventional stereo matching methods [Olsson *et al.*, 2013; Wang and Yang, 2011; Kolmogorov and Zabih, 2002, 2001; Boykov *et al.*, 2001]. In the MRF framework, each pixel $p \in (\mathcal{P} \subset \mathbb{Z}^2)$ is assigned a value in some disparity space $\mathcal{S}$, and one seeks a disparity map f for every pixel $f_p = f(p) : \mathcal{P} \rightarrow \mathcal{S}$ that minimizes

$$E(f) = \sum_{p \in \mathcal{P}} \phi_p(f_p) + \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{N}(p)} \psi_{pq}(f_p, f_q). \tag{3.2}$$

The first term, called the *data term* or *unary term*, measures the photo-consistency between matching pixels. The disparity f_p defines a warp from a pixel p in one image to its correspondence in the other image. The second term is called the *smoothness term* or *pairwise term*, which penalizes discontinuity of disparities of a pixel p and its neighboring pixels $q \in \mathcal{N}(p)$. We define these terms as below.

Data Term

To measure photo-consistencies, we use a data term that has been recently proposed by [Bleyer *et al.*, 2011]. Here, each pixel p's disparity d_p is over-parameterized by a 3D plane $d_p = a_p x + b_p y + c_p$ to avoid the frontal-parallel bias. Therefore, the objective becomes to seek a disparity plane $f_p = (a_p, b_p, c_p)^T \in \mathcal{S}$ for every pixel p in the left and right images such that disparity map f minimizes the energy function $E(f)$ of Equation (3.2). Using this p's disparity plane f_p, a pixel $q = (q_x, q_y)^T$ in the left image is warped to a new location in the right image by a warping w_{f_p} as

$$w_{f_p}(q) = q - (a_p q_x + b_p q_y + c_p, 0)^T. \tag{3.3}$$

The data term of p in the left image is therefore defined as

$$\phi_p(f_p) = \sum_{q \in W_p} \omega_{pq}\, \rho\left(q, w_{f_p}(q)\right). \tag{3.4}$$

Here, W_p is a square window centered at p. The weight ω_{pq} implements the adaptive support window proposed in [Yoon and Kweon, 2005], and is defined as

$$\omega_{pq} = e^{-\|I_L(p) - I_L(q)\|_1 / \gamma}, \tag{3.5}$$

where γ is a user-defined parameter, and $\|\cdot\|_1$ represents the ℓ_1-norm. Unlike the original weight function, the spatial distance term is removed because it makes only a slight contribution to improving the results as stated in [Hosni *et al.*, 2012; Bleyer *et al.*, 2011]. Our color-based weights ω_{pq} are illustrated in Figure 3.3 (d). The function $\rho(q, w_{f_p}(q))$ measures the pixel dissimilarity between q and its matching point $w_{f_p}(q)$ as

$$\begin{aligned} \rho(q, w_{f_p}(q)) = (1-\alpha)\ &\min(\|I_L(q) - I_R(w_{f_p}(q))\|_1, \tau_{col}) \\ +\alpha\ &\min(\|\nabla_x I_L(q) - \nabla_x I_R(w_{f_p}(q))\|_1, \tau_{grad}), \end{aligned} \tag{3.6}$$

where $\nabla_x I$ represents the x-component of the gray-value gradient of image I, and α is a factor that balances the weights of color and gradient terms. The two terms are truncated by τ_{col} and τ_{grad} to increase the robustness for occluded regions. We use linear interpolation to compute $I_R(w_{f_p}(q))$. When the data term is defined on the right image, we swap I_L and I_R in Equations (3.5) and (3.6), and add the disparity value in Equation (3.3).

Smoothness Term

For the smoothness term, we use a curvature-based, second-order smooth regularization term [Olsson *et al.*, 2013] defined as

$$\psi_{pq}(f_p, f_q) = \lambda \max\left(\omega_{pq}, \epsilon\right)\ \min(\bar{\psi}_{pq}(f_p, f_q), \tau_{dis}), \tag{3.7}$$

where λ is a user-defined parameter, and ϵ is a small user-defined value that gives a lower bound to the weight ω_{pq} for increasing the robustness. The function $\bar{\psi}_{pq}(f_p, f_q)$ penalizes the discontinuity between f_p and f_q in terms of disparity as

$$\bar{\psi}_{pq}(f_p, f_q) = |d_p(f_p) - d_p(f_q)| + |d_q(f_q) - d_q(f_p)|, \tag{3.8}$$

where $d_p(f_q) = a_q p_x + b_q p_y + c_q$. The deviations $|d_p(f_p) - d_p(f_q)|$ and $|d_q(f_q) - d_q(f_p)|$ in Equation (3.8) are illustrated as red arrows in Figure 3.4 (a). The sum of the deviations $\bar{\psi}_{pq}(f_p, f_q)$ is truncated by τ_{dis} to allow sharp jumps in disparity at depth edges. This formulation enforces second order smoothness for the resulting disparity map, because it gives no penalty when p and q are on the same disparity plane as shown in Figure 3.4 (b). Note that if $a = b = 0$ (front-parallel plane) is forced as shown in Figure 3.4 (c), this smoothness function becomes the standard truncated linear model $\min(2|d_p - d_q|, \tau_{dis})$ where $d_p = c_p$ is a scalar disparity value assigned to pixel p. This term has a front-parallel bias and should be avoided [Woodford *et al.*, 2009; Olsson *et al.*, 2013]. Notice that in spite of its useful property, our truncated smoothness term cannot be used in the convex optimization framework of [Heise *et al.*, 2013] for its high non-convexity.

Our smoothness term is submodular under expansion moves for taking advantage of GC. A proof is given by [Olsson *et al.*, 2013] but for the completeness we re-state it as the following lemma.

Lemma A: *The term $\psi_{pq}(f_p, f_q)$ defined by Equation (3.7) satisfies the following submodularity of expansion moves [Boykov* et al.*, 2001]:*

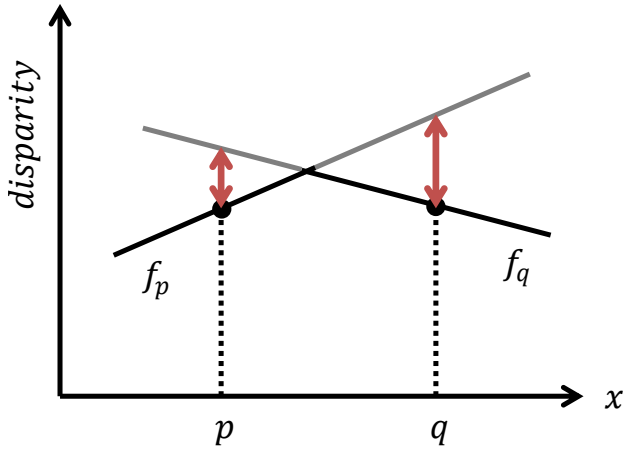
$$\psi_{pq}(\alpha, \alpha) + \psi_{pq}(\beta, \gamma) \leq \psi_{pq}(\beta, \alpha) + \psi_{pq}(\alpha, \gamma). \tag{3.9}$$

Proof. Obviously, $\bar{\psi}_{pq}(\alpha, \alpha) = 0$. Therefore,

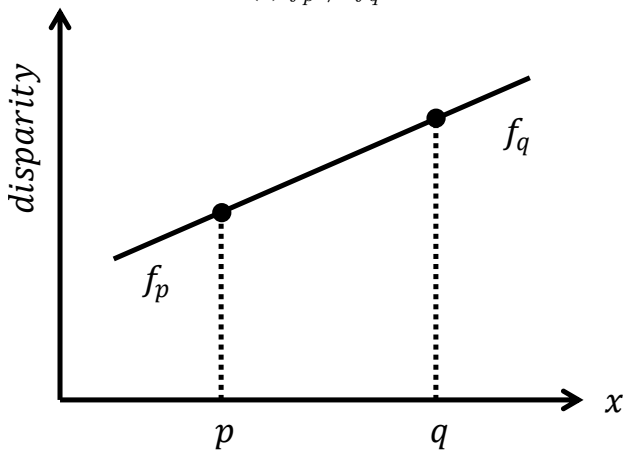
$$\begin{aligned}
\bar{\psi}_{pq}(\alpha, \alpha) + \bar{\psi}_{pq}(\beta, \gamma) &= \bar{\psi}_{pq}(\beta, \gamma) \\
&= |d_p(\beta) - d_p(\gamma)| + |d_q(\beta) - d_q(\gamma)| \\
&= |\,(d_p(\beta) - d_p(\alpha)) - (d_p(\gamma) - d_p(\alpha))\,| + |\,(d_q(\beta) - d_q(\alpha)) - (d_q(\gamma) - d_q(\alpha))\,| \\
&\leq |d_p(\beta) - d_p(\alpha)| + |d_p(\gamma) - d_p(\alpha)| + |d_q(\beta) - d_q(\alpha)| + |d_q(\gamma) - d_q(\alpha)| \\
&= \bar{\psi}_{pq}(\beta, \alpha) + \bar{\psi}_{pq}(\alpha, \gamma).
\end{aligned} \tag{3.10}$$

Thus, $\bar{\psi}_{pq}(f_p, f_q)$ satisfies the submodularity of expansion moves. For its truncated function,

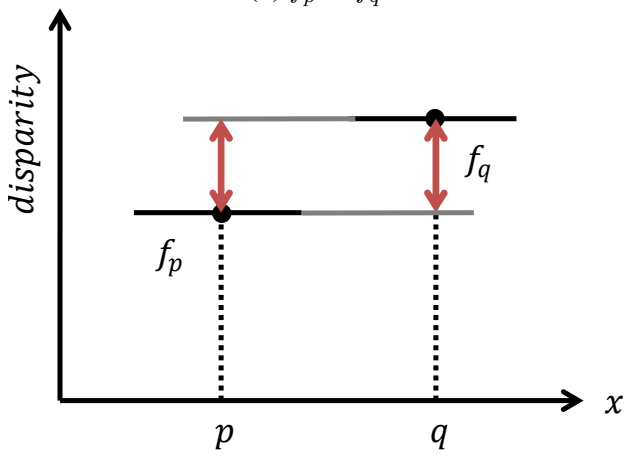
$$\begin{aligned}
\min\left(\bar{\psi}_{pq}(\alpha, \alpha), \tau\right) + \min\left(\bar{\psi}_{pq}(\beta, \gamma), \tau\right) &= \min\left(\bar{\psi}_{pq}(\beta, \gamma), \tau\right) \\
&\leq \min\left(\bar{\psi}_{pq}(\beta, \alpha) + \bar{\psi}_{pq}(\alpha, \gamma), \tau\right) \\
&\leq \min\left(\bar{\psi}_{pq}(\beta, \alpha), \tau\right) + \min\left(\bar{\psi}_{pq}(\alpha, \gamma), \tau\right).
\end{aligned} \tag{3.11}$$



(a) $f_p \neq f_q$



(b) $f_p = f_q$



(c) $a = b = 0$

Figure 3.4 Illustration of our smoothness term. Essentially, our smoothness term penalizes the deviations of neighboring disparity planes shown as red arrows in (a). When neighboring pixels are on the same disparity plane as shown in (b), the smoothness term gives no penalty; thus, it enforces second order smoothness for the resulting disparity map. If $a = b = 0$ (front-parallel plane) is forced as shown in (c), the term becomes equivalent to the standard truncated linear model, yielding a front-parallel bias.

Therefore, the truncated function of $\bar{\psi}_{pq}(f_p, f_q)$ also satisfies the submodularity. Because $\psi_{pq}(f_p, f_q)$ is a constant-factored function of $\bar{\psi}_{pq}(f_p, f_q)$, *i.e.*, $\lambda \max(\omega_{pq}, \epsilon)$ is constant to f, our smoothness term $\psi_{pq}(f_p, f_q)$ also satisfies the submodularity. □

3.3.2 Locally Shared Labels

As the main contribution of this paper, we introduce locally shared labels for efficiently optimizing continuous MRFs. The locally shared labels are the combination of pixel and region labels, in which label spaces are *shared* among neighbors, and they enable per-pixel estimation of continuous solutions as well as fast propagations.

Pixel and Region Labels

Pixel labels are a small number (say, K) of discrete disparity labels (or candidate labels) defined at each pixel p, which we refer to as a *pixel label set*, $L_p = \{l_p^{(0)}, l_p^{(1)}, \ldots, l_p^{(K-1)}\}$, $l_p^{(i)} \in \mathcal{S}$. The pixel label sets are shared among neighboring pixels. In addition, we define region labels that give additional candidate labels for accelerating spatial propagation and avoiding stuck at a local minima. We use a regular grid structure for regions, which are indexed by the region coordinates $r \in (\mathcal{R} \subset \mathbb{Z}^2)$ like the pixel coordinates. Region labels define for a region r a set of K_R candidate labels $R_r \subset \mathcal{S}$, which we call a *region label set*. Each label set R_r gives candidate labels for pixels in the region r and pixels in the neighboring regions as well, *i.e.*, R_r is also shared among neighboring regions just like pixel labels.

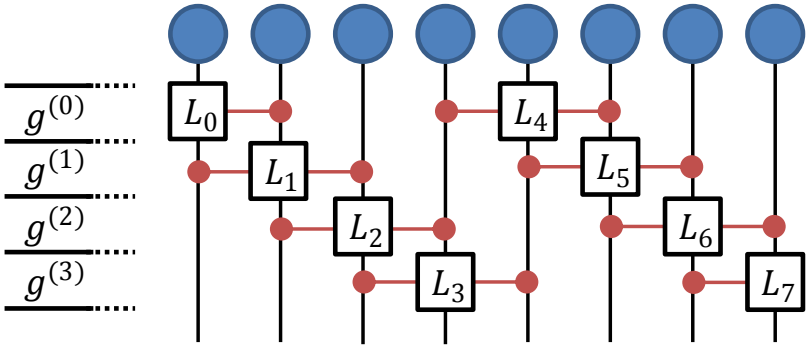
During the inference, for each pixel p, our method chooses the best candidate label f_p from the union of pixel and region label sets that are shared for the pixel p:

$$C_p = L_p \cup L_q \cup R_r \cup R_s, \tag{3.12}$$

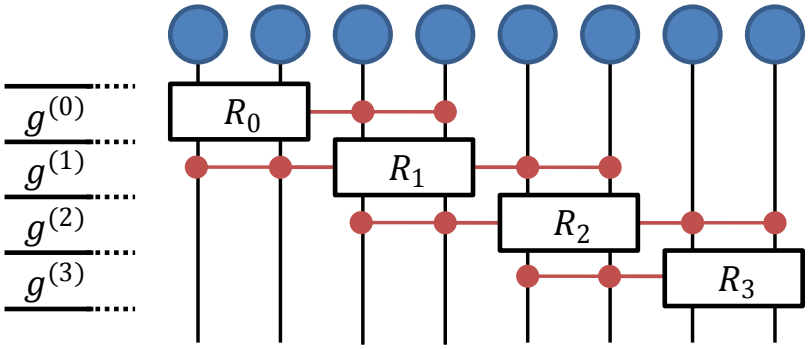
where q, r, s represent p's neighboring pixels, the region that p belongs to, and the neighboring regions to r, respectively. By sharing label sets among neighbors, good candidate labels are spatially propagated to nearby pixels. The concept of pixel and region labels is illustrated in Figure 3.5.

Proposal Generation for Fusion

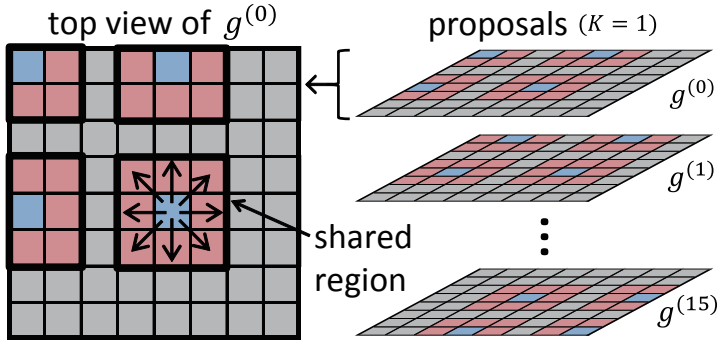
During the inference, we repeatedly seek the best labeling $f^{(t)}$ for the current label sets $\{L_p\}$ and $\{R_r\}$, and refine the label sets. The former part, *i.e.*, the selection and propagation of candidate labels in $\{L_p\}$ and $\{R_r\}$, is cast as fusion-based energy minimization [Lempitsky *et al.*, 2010], as described in the rest of this section. Consider the essential function of fusion is to make a good solution f by fusing a number of proposal disparity maps $\{g^{(0)}, g^{(1)}, \ldots, g^{(n-1)}\}$, where at each pixel f_p is assigned one of n disparity labels $\{g_p^{(0)}, g_p^{(1)}, \ldots, g_p^{(n-1)}\}$. In our method, we build a special form of proposals from $\{L_p\}$ and $\{R_r\}$ in a manner that achieves the propagation of pixel



$(K = 1)$ (a) pixel labels (1D image)



$(K = 1)$ (b) region labels (1D image)



(c) proposal construction in case of pixel labels

Figure 3.5 Illustrations of pixel and region labels, and proposal construction. For simplicity, they are illustrated by 1D images with the blue nodes representing pixels. The boxes L_p and R_r represent a set of candidate disparity labels given for the pixel p and pixels in the region r, respectively. The horizontal red lines signify that the label sets $\{L_p\}$ and $\{R_r\}$ are shared between neighbors. The label sets are aligned so as to make proposal disparity maps $g^{(j)}$ for fusion. In particular, proposal construction of pixel labels is illustrated for a 2D image in (c) indicating that, in each proposal $g^{(j)}$, candidate labels defined at blue pixels are shared between red neighbors.

and region labels. To make proposals from pixel labels, we copy each candidate label $l_p^{(i)} \in L_p$ to the j-th proposal $g^{(j)}$ by setting

$$g_q^{(j)} \leftarrow l_p^{(i)}, \tag{3.13}$$

where q is the nine neighboring pixels around and including p (*i.e.*, the pixels where the candidate label $l_p^{(i)}$ is shared), and j is given as

$$j = K\ (4\,(p_y \bmod 4) + (p_x \bmod 4)) + i. \tag{3.14}$$

Here, mod is the modulo operation, and p_x and p_y are p's coordinates specified as $p_x \in [0, \text{width} - 1]$ and $p_y \in [0, \text{height} - 1]$. Figure 3.5 (a) illustrates this construction for the case of a 1D image with $K = 1$ (*i.e.*, $L_p = \{l_p^{(0)}\}$) for simplicity, where a horizontal layer of candidate labels at vertical position j represents a proposal $g^{(j)}$. Figure 3.5 (c) illustrates for a 2D image showing that candidate labels at blue pixels are shared among red neighbors in each proposal $g^{(j)}$. The integer 4 in Equation (3.14) means that, in each proposal, we leave a "gap" (shown as gray pixels in Figure 3.5 (c) that represent no candidate labels) between each "shared region" (see Figure 3.5 (c)) for ensuring submodularity, which we describe later. We assign an infinite unary cost to those invalid labels to ensure that such labels are avoided during the inference. For region labels, proposals are constructed in the same manner with pixel labels by regarding a region as a pixel as shown in Figure 3.5 (b). The fusion is performed using the proposals generated from both pixel and region labels. The visualization of this process with real data is shown in Figure 3.6.

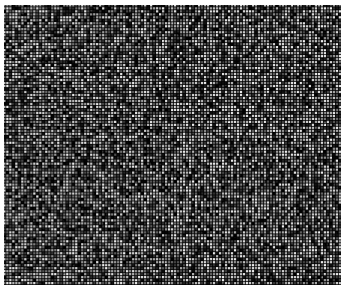
This particular form of proposal construction guarantees that a binary fusion of an arbitrary solution f and any of the proposals $g = g^{(j)}$ is submodular, thus it is *exactly* solved via GC. A proof and detailed descriptions about this submodularity are given in the next section. With this submodularity guarantee, we only need to use a standard GC [Kolmogorov and Zabin, 2004; Boykov and Kolmogorov, 2004] instead of employing expensive QPBO-GC [Kolmogorov and Rother, 2007] used in usual fusion moves [Lempitsky *et al.*, 2010].

In addition, this proposal generation helps obtain smooth solutions because multiple pixels in shared regions are allowed to move-at-once to the same candidate label at one binary fusion. This effect becomes more significant with region labels because of their large shared regions. In fact, region labels make the key factor in our algorithm for both efficiency and accuracy as we will see in the experiment.

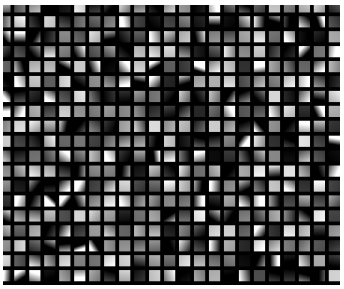
Submodular Fusion Moves with Locally Shared Labels

Here we prove the following statement:

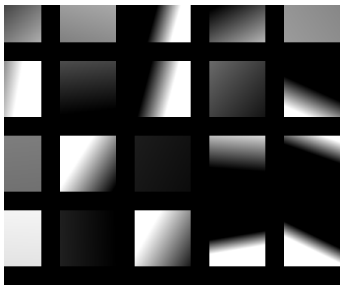
Lemma B: *If g is a proposal solution constructed from pixel or region labels, the binary-fusion energy $E(f'|f,g)$ is submodular,* i.e.*, all pairwise terms ψ_{pq} in $E(f'|f,g)$ satisfy the following submodularity of fusion moves [Lempitsky* et al.*, 2010; Kolmogorov and Rother, 2007; Kolmogorov*



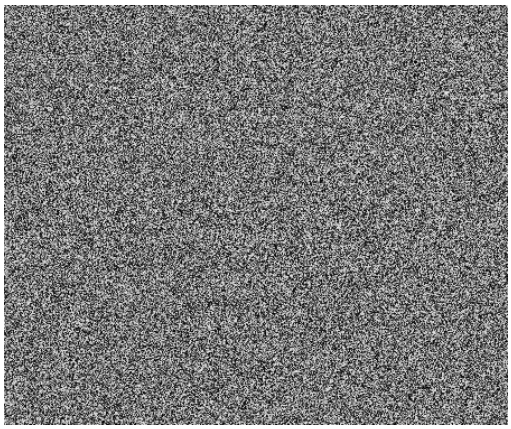
(a) pixel labels



(b) region labels of 5×5-size

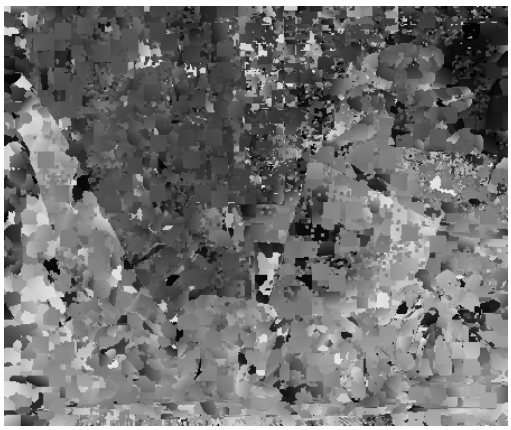


(c) region labels of 25×25-size

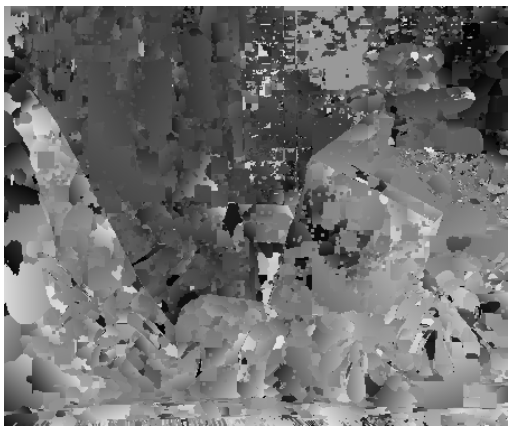


(d) initial random disparity map

(e) after fused with pixel labels

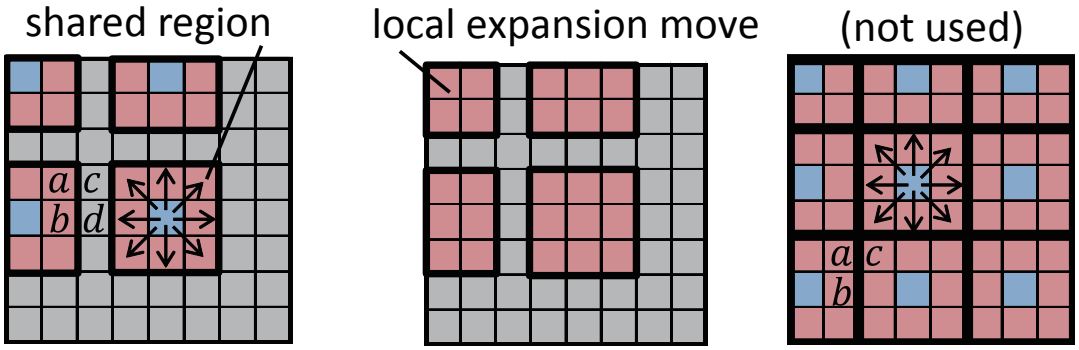


(f) after fused with region labels of 5×5-size



(g) after fused with region labels of 25×25-size

Figure 3.6 Fusion process with real data. We show intermediate fusion process in the first iteration for Teddy, *i.e.*, the process from Figure 3.1 (a) to (b). Here, (d) an initial random disparity map is successively fused with proposals generated from (a) pixel labels, (b) region labels of 5×5-size, and (c) region labels of 25×25-size, resulting in intermediate disparity maps shown in (e), (f), and (g), respectively.



(a) Proposal from pixe labels

(b) Multiple local-expansion-moves

(c) Non-submodular proposal

Figure 3.7 Illustration of our proposals using local shared labels. An example proposal generated from pixel labels is shown in (a). Because expansion moves are fusion with globally-constant proposals, our locally-constant proposals can be regarded as multiple local-expansion-moves as shown in (b). If proposals are made so that share regions are densely aligned as shown in (c), this will produce non-submodular terms. So we do not use such proposals.

and Zabin, 2004]:

$$\psi_{pq}(g_p, g_q) + \psi_{pq}(f_p, f_q) \leq \psi_{pq}(f_p, g_q) + \psi_{pq}(g_p, f_q). \tag{3.15}$$

If it holds, the binary-fusion energy $E(f'|f, g)$ can be optimally minimized via GC. Note that this condition does not generally hold in standard fusion moves [Lempitsky *et al.*, 2010] that assume g is arbitrary.

Proof. We show here only the case of g generated from pixel labels, but the same result can be easily derived for region labels. See Figure 3.7 (a) that depicts an example proposal g constructed from pixel labels. We make g by setting a consistent label at pixels in each "shared region", and an invalid label with an infinite unary cost at gray pixels. Here, all the pairwise terms $\psi_{pq}(f'_p, f'_q)$ in $E(f'|f, g)$ can be summarized into three types: pairwise terms inside shared regions $\psi_{ab}(f'_a, f'_b)$, outside shared regions $\psi_{cd}(f'_c, f'_d)$, and between shared and invalid regions $\psi_{ac}(f'_a, f'_c)$. Examples of the four pixels a, b, c, and d are visualized in Figure 3.7 (a). Because of the infinite unary costs of g_c and g_d, the binary variables f'_c and f'_d are forced to take their current labels f_c and f_d, respectively. Thus, $\psi_{ac}(f'_a, f'_c)$ becomes an a's unary potential $\psi_{ac}(f'_a, f_c)$, and $\psi_{cd}(f'_c, f'_d)$ becomes a constant energy $\psi_{cd}(f_c, f_d)$, both of which can be ignored from the submodularity condition that only depends on pairwise potentials. The remaining pairwise terms are those inside shared regions $\psi_{ab}(f'_a, f'_b)$, for which $g_a = g_b$ holds because labels of g are constant in each shared region. Using these results, we substitute $g_p = g_q = \alpha$, $f_p = \beta$, and $f_q = \gamma$ into Equation (3.15), resulting in a relaxed condition known as the submodularity of expansion moves shown in above Equation (3.9). This condition holds for our pairwise term ψ_{pq}, as proved in the previous section. □

To understand this result intuitively, we remind readers that expansion moves [Boykov *et al.*, 2001] are special cases of fusion moves where each proposal g is given globally-constant, *i.e.*, g_p takes a consistent label for all pixels p. On the other hand, our proposal g is made *locally-constant* by shared regions; thus, a fusion move with our proposal g can be interpreted as multiple local-expansion-moves, where each shared region produces one local-expansion-move, as illustrated in Figure 3.7 (b). Therefore, in our method, the condition for deriving submodularity is the same with that of expansion moves.

Note that if we use an integer 3 instead of 4 in Equation (3.14) that we use for generating proposals $g^{(j)}$ from pixel labels, we obtain proposals where shared regions are densely aligned without leaving "gaps" of invalid labels, as illustrated in Figure 3.7 (c). In this case, $E(f'|f,g)$ is not generally submodular. To see this, we summarize all pairwise terms in $E(f'|f,g)$ into two types: those inside the same shared regions $\psi_{ab}(f'_a, f'_b)$, and between two shared regions $\psi_{ac}(f'_a, f'_c)$. Examples of the three pixels a, b, and c are visualized in Figure 3.7 (c). Here, the terms $\psi_{ac}(f'_a, f'_c)$ can be non-submodular because all of g_a, g_c, f_a, and f_c are arbitrary.

3.3.3 Optimization

The overview of our optimization procedure is summarized in Algorithm 3.1.

Algorithm 3.1 Overview of the proposed optimization procedure

Initialize $\{L_p\}$ and $\{R_r\}$ randomly.
repeat
  $\diamondsuit$ **Optimize labeling** f **for current label sets**:
  $f^{(t)} = \text{argmin}\, E(f)$ with label sets $\{L_p\}$ and $\{R_r\}$
  $\diamondsuit$ **Refine label sets** $\{L_p\}$ **and** $\{R_r\}$:
  for all pixels $p \in \mathcal{P}$ **do**
    $\tilde{C}_p \leftarrow C_p$ with perturbation.
    $L_p \leftarrow$ best $K-1$ candidate labels $c \in (L_p \cup \tilde{C}_p) \setminus \{f_p^{(t)}\}$ that minimize $E_p(c|f^{(t)})$
    $L_p \leftarrow L_p \cup \{f_p^{(t)}\}$
  end for
  for all regions $r \in \mathcal{R}$ **do**
    $R_r \leftarrow$ random K_R candidate labels from $\{f_p^{(t)} |$ pixels p in the region $r\}$
  end for
until convergence

As discussed in the previous section, our optimization uses an iterative framework, where we alternately optimize the labeling f with given label sets $\{L_p\}$ and $\{R_r\}$, and refine the label sets $\{L_p\}$ and $\{R_r\}$ locally with the labeling f fixed. This alternating optimization is similar to the PEARL algorithm [Isack and Boykov, 2012] described in Section 2.4.2.

Our optimization begins with randomly initializing the label sets $\{L_p\}$ and $\{R_r\}$. To sample the allowed solution space evenly, we take the initialization strategy described in [Bleyer *et al.*, 2011]. For $l_p^{(i)} \in L_p$ at $p = (p_x, p_y)^T$, we select a random disparity z_0 in the allowed disparity range

$[0, \text{dispmax}]$. Then, a random unit vector $n=(n_x, n_y, n_z)^T$ and z_0 are converted to the plane representation by $a_p = -n_x/n_z$, $b_p = -n_y/n_z$, and $c_p = -(n_x p_x + n_y p_y + n_z z_0)/n_z$. For the region label sets $\{R_r\}$, we randomly pick K_R pixels in each region, and copy the candidate label $l_p^{(0)} \in L_p$ of the randomly chosen pixels p to the region label set.

At line 3 of Algorithm 3.1, we optimize the labeling f by the procedure illustrated in the previous section. It can be approximately solved by fusing proposals constructed from pixel and region label sets $\{L_p\}$ and $\{R_r\}$. This process is visualized in Figure 3.6

In lines 5–9, we refine the pixel label sets $\{L_p\}$. At each pixel p, we first randomly perturb p's candidate labels C_p of Equation (3.12) and obtain $\tilde{C}_p$. As the refined L_p, we select the best K candidate labels from the union of $\tilde{C}_p$ and the current L_p that minimize the following local energy at the pixel p:

$$E_p(s|f^{(t)}) = \phi_p(s) + \sum_{q \in \mathcal{N}(p)} \psi_{pq}(s, f_q^{(t)}) \ , \ \ s \in \mathcal{S}. \tag{3.16}$$

Here, the refined label set L_p is forced to contain the current candidate label $f_p^{(t)}$ to ensure that, in the next iteration, the solution $f^{(t+1)}$ can stay at $f^{(t)}$, thereby the energy does not increase, *i.e.*, $E(f^{(t)}) \geq E(f^{(t+1)})$ holds throughout the iterations. Perturbation is implemented as described in [Bleyer *et al.*, 2011]. Namely, each candidate label $(a, b, c)^T \in C_p$ is converted to disparity d and normal vector n. We then add a random disparity $\Delta_d \in [-r_d, r_d]$ and a random unit vector Δ_n to them, respectively, as $d' = d_p + \Delta_d$ and $n' = n + r_n \Delta_n$. Finally, d' and $n'/|n'|$ are converted to the plane representation $(a', b', c')^T \in \tilde{C}_p$ as a perturbed candidate label. The values r_d and r_n define an allowed change of planes. We start by setting $r_d \leftarrow \text{dispmax}/2$ and $r_n \leftarrow 1$. After each iteration, we update them by $r_d \leftarrow r_d/2$ and $r_n \leftarrow r_n/2$.

In lines 10–12, we update the region label sets $\{R_r\}$. As done in the initialization, we again take a random-pick-up scheme. This time, the current solution $f_p^{(t)}$ of randomly chosen pixels p is taken as the region labels.

Finally, after the whole process, we perform the post-processing using left-right consistency check and median filtering as described in [Bleyer *et al.*, 2011] for further improving the results. This step is mainly for estimating disparity at occluded pixels but some mismatching pixels are also improved. This scheme is widely employed in recent methods [Rhemann *et al.*, 2011; Bleyer *et al.*, 2011; Besse *et al.*, 2012; Lu *et al.*, 2013; Heise *et al.*, 2013].

3.4 Experiments

In the experiments, we first evaluate our method on the Middlebury benchmark. Then, we assess the effect of region labels, and compare our method with PMBP [Besse *et al.*, 2012] that is closely related to our approach. To further demonstrate the effectiveness of our method, we show additional results for two outdoor scenes and nine Middlebury datasets including failure examples.

Settings

We use the following settings as default throughout the experiments. We use a PC with a Xeon CPU (2.53 GHz $\times$ 4 cores) and NVIDIA GeForce GTX-295 GPU[3]. The parameters of our data term are set as $\{\tau_{col}, \tau_{grad}, \gamma, \alpha\} = \{10, 2, 10, 0.9\}$ as specified in [Bleyer *et al.*, 2011]. The size of supporting windows is set to 41×41, which is the same setting with PMBP [Besse *et al.*, 2012]. For the smoothness term, we use $\{\lambda, \tau_{dis}, \epsilon\} = \{20, 1, 0.01\}$ and eight neighbors for $\mathcal{N}$. For optimization, three-layer locally shared labels are used: pixel labels with $K = 2$, region labels of size 5×5 with $K_R = 2$, and also regions labels of size 25×25 with $K_R = 2$, and a GC implementation of [Boykov and Kolmogorov, 2004] is used. We iterate twice for each proposal in fusion stage, and iterate the outer-loop process ten times. The computation of unary costs is performed in parallel on GPU, and pairwise costs are computed on four CPU cores.

3.4.1 Evaluation on the Middlebury Benchmark

We show in Table 3.2 selected rankings on the Middlebury stereo benchmark for 0.5-pixel accuracy. Our method achieves the current best average rank (3.5) and bad-pixel-rate (6.63%) amongst more than 155 stereo methods. Even without post-processing, our method still outperforms the other methods in average rank, despite that methods [Bleyer *et al.*, 2011; Besse *et al.*, 2012; Lu *et al.*, 2013; Heise *et al.*, 2013] use the post-processing. Compared with closely related approaches (PMBP [Besse *et al.*, 2012] and PatchMatch stereo [Bleyer *et al.*, 2011]), which are ranked seventh and ninth in Table 3.2, although results of PMBP for Cones are slightly better than ours, our method consistently outperforms the two methods in the other evaluations. We summarize the results of our method in Figures 3.8 to 3.11. Note that Tsukuba may not be appropriate for accurate evaluations, because its ground truth has only integer precision. More results including outdoor scenes and failures are shown in Sections 3.4.4 and 3.4.5.

3.4.2 Effect of Region Labels

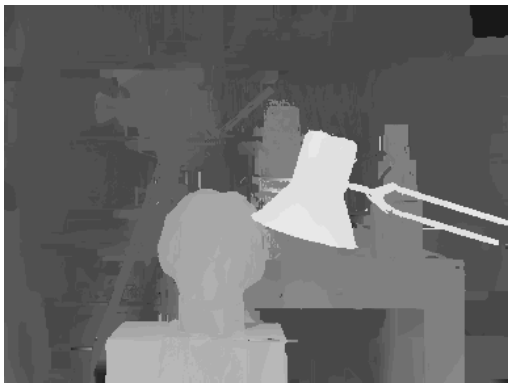
To observe the effect of region labels, we assess the performance using three different settings: (1) only pixel labels with $K = 6$; (2) pixel labels with $K = 4$ and region labels of size 5×5 with $K_R = 2$; (3) pixel labels with $K = 2$, region labels of size 5×5 with $K_R = 2$, and also regions labels of size 25×25 with $K_R = 2$ (the default setting for our method described above). Among the three settings, the number of candidate labels given for each pixel (*i.e.*, $|C_p|$) is kept consistent. We use $\lambda = 40$ but keep the other parameters as default. Using these settings, we observe the performance variations by estimating the disparities of only the left image of the Cloth1 dataset without the post-processing.

In Figure 3.12, we show the results of the three cases after ten iterations. Also, plots in Figures 3.13 to 3.15 show the energy variations of the energy function, data term, and smoothness

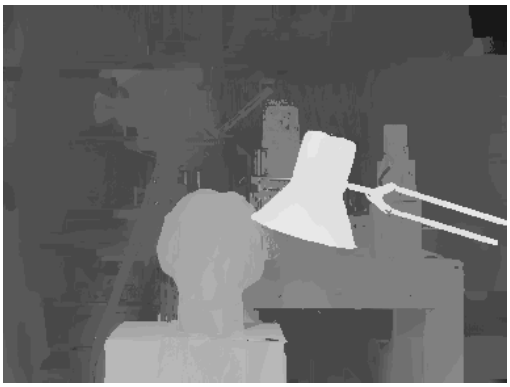
[3] Although GTX-295 has two GPU cores, in this experiment we used only one of them. Our method itself allows multi-core GPU processing.

(a) left view

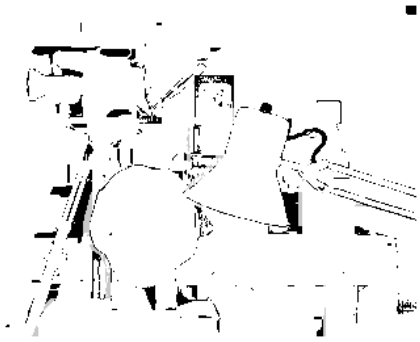
(b) right view



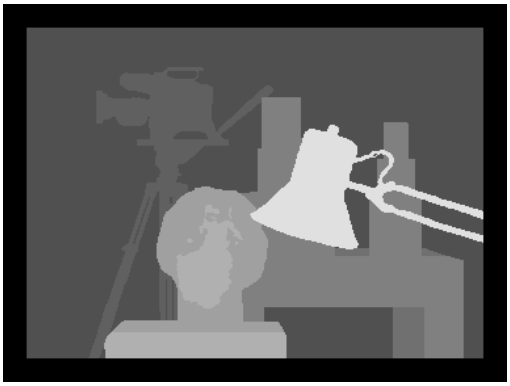
(c) raw result



(d) after post-processing

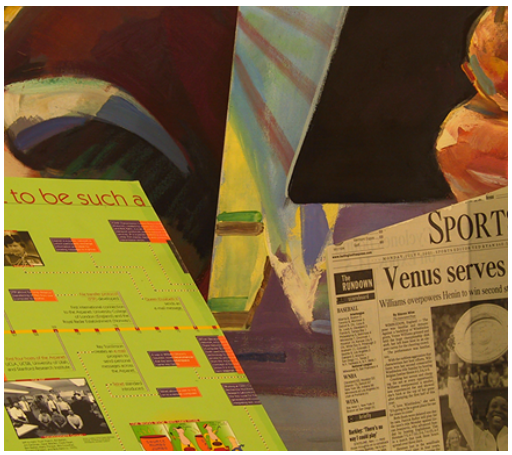


(e) error map (0.5-pixel threshold)



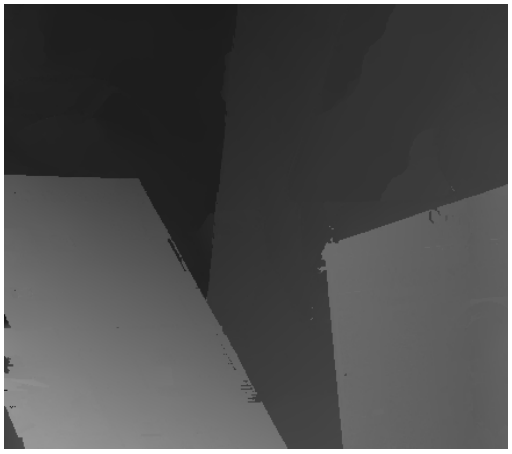
(f) ground truth

Figure 3.8 Results of Venus in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result before post-processing, (d) after post-processing, (e) the error map of the result after post-processing, and (f) the ground truth. In the error maps, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels. Note that Tsukuba may not be appropriate for accurate evaluations because its ground truth has only integer precision.

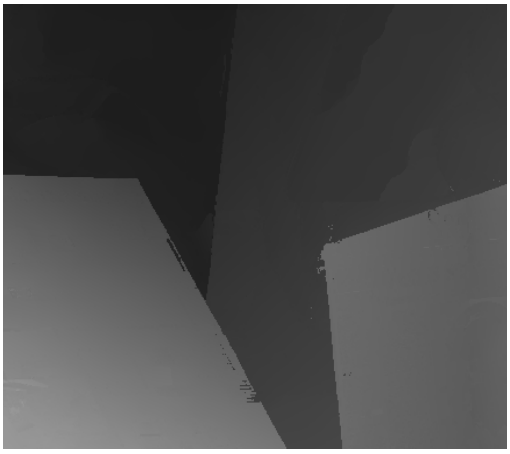


(a) left view

(b) right view

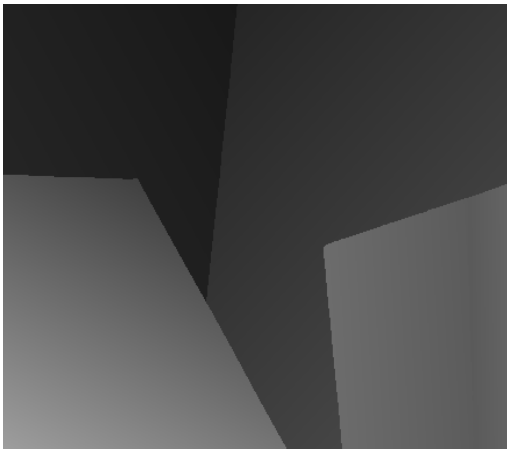


(c) raw result



(d) after post-processing

(e) error map (0.5-pixel threshold)

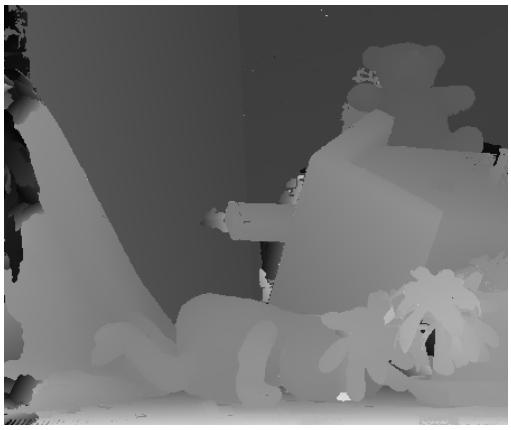


(f) ground truth

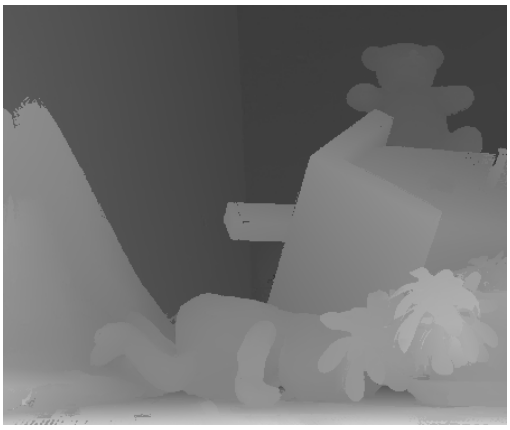
Figure 3.9 Results of Venus in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result before post-processing, (d) after post-processing, (e) the error map of the result after post-processing, and (f) the ground truth. In the error maps, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.

(a) left view

(b) right view

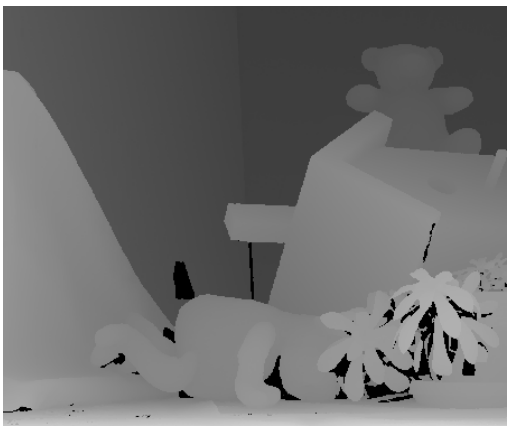


(c) raw result



(d) after post-processing

(e) error map (0.5-pixel threshold)

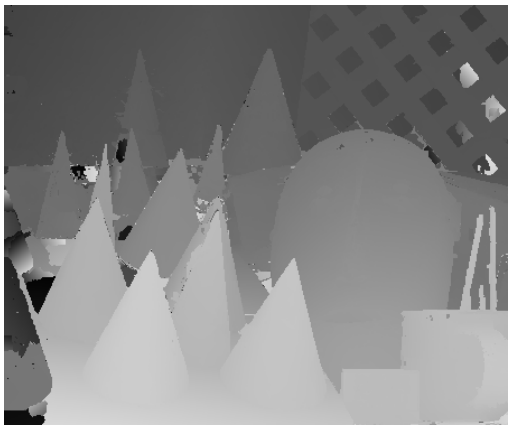


(f) ground truth

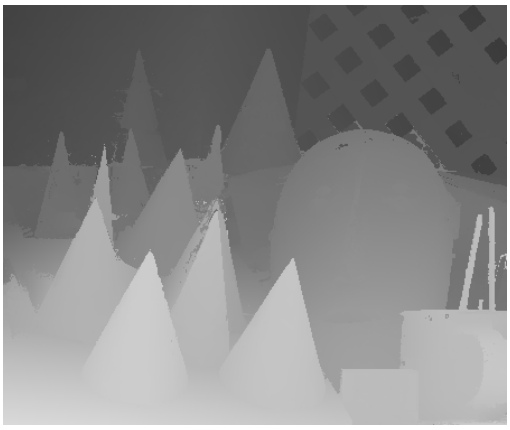
Figure 3.10 Results of Teddy in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result before post-processing, (d) after post-processing, (e) the error map of the result after post-processing, and (f) the ground truth. In the error maps, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.

(a) left view

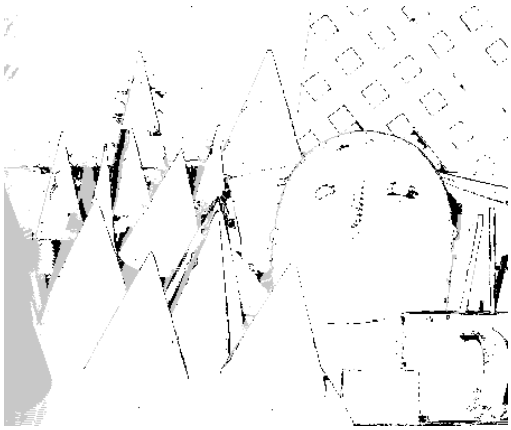
(b) right view



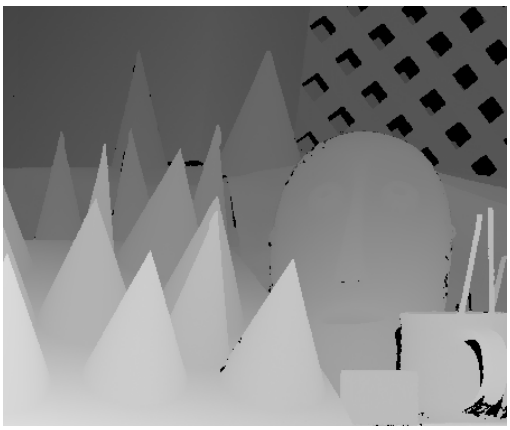
(c) raw result



(d) after post-processing



(e) error map (0.5-pixel threshold)



(f) ground truth

Figure 3.11 Results of Cones in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result before post-processing, (d) after post-processing, (e) the error map of the result after post-processing, and (f) the ground truth. In the error map, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.

Table 3.2 Middlebury benchmark evaluations for 0.5-pixel precision. Our method achieves the current best average rank of 3.5 and bad-pixel-rate of 6.63% among more than 150 stereo algorithms. In the last row, we also show the results by our method without post-processing, which still outperform the other methods in average rank and bad-pixel-rate. In *all*, results are evaluated for all pixels where the ground truth is given, while only for non-occluded pixels in *nonocc*, and around depth discontinuities in *disc*. PM-Huber [Heise *et al.*, 2013] uses Huber regularization in a joint framework of PatchMatch and convex optimization, thus their approach is limited to convex energies. SubPixSearch [Mizukami *et al.*, 2012] finds sub-pixel disparity by refining integer-valued initial disparity maps. PMF [Lu *et al.*, 2013] incorporates fast ege-preserving filtering techniques into PatchMatch stereo for reducing computational complexity, but uses no smoothness regularization. PMBP [Besse *et al.*, 2012] incorporates smoothness regularization into PatchMatch stereo using BP. PM (PatchMatch stereo) [Bleyer *et al.*, 2011] uses PatchMatch inference for estimating per-pixel disparity planes but uses no smoothness regularization. The rankings are obtained on November first, 2013 at the online benchmark site of [Scharstein and Szeliski, 2001].

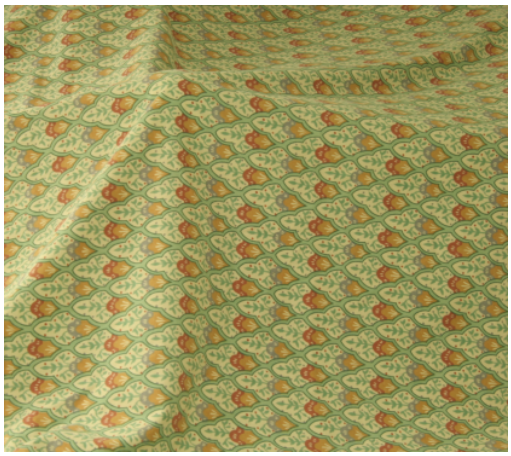
Algorithm	Avg. Rank	Tsukuba nonocc	Tsukuba all	Tsukuba disc	Venus nonocc	Venus all	Venus disc	Teddy nonocc	Teddy all	Teddy disc	Cones nonocc	Cones all	Cones disc	Average Percent Bad Pixels
1. OUR METHOD	3.5	5.04 2	5.56 2	14.0 9	0.66 2	0.88 2	5.82 4	**4.20** 1	**7.12** 1	**12.9** 1	3.77 5	9.16 5	10.4 8	6.63
2. PM-Huber	5.4	7.12 9	7.80 8	13.7 7	1.00 8	1.40 9	7.80 12	5.53 3	9.36 2	15.9 4	**2.70** 1	**7.90** 1	**7.77** 1	7.33
3. SubPixSearch	6.2	5.60 3	6.23 3	9.46 3	1.07 10	1.64 10	7.36 8	6.71 6	11.0 4	16.9 6	4.02 8	9.76 6	10.3 7	7.51
4. PMF	8.6	11.0 29	11.4 26	16.0 24	0.72 4	0.92 3	5.27 3	4.45 2	9.44 3	13.7 2	2.89 2	8.31 3	8.22 2	7.69
7. PMBP	13.2	11.9 40	12.3 36	17.8 43	0.85 6	1.10 4	6.45 6	5.60 4	12.0 6	15.5 3	3.48 3	8.88 4	9.41 4	8.77
9. PatchMatch	20.3	15.0 57	15.4 56	20.3 69	1.00 9	1.34 8	7.75 11	5.66 5	11.8 5	16.5 5	3.80 6	10.2 7	10.2 6	9.91
***. w/o post-proc.**	4.3	5.15 2	5.82 2	14.0 9	0.73 4	1.02 3	6.65 6	4.65 2	10.8 3	14.3 2	3.88 6	9.71 5	10.7 8	7.29

term after each iteration, respectively. Figure 3.13 shows that region labels play a critical role in minimizing energies. Figures 3.14 and 3.15 indicate that region labels effectively reduce the energies of both the data and smoothness terms, which shows the contribution of region labels for fast propagation of good candidate labels as we intended. On the other hand, if only pixel labels are used, the solution is trapped at a bad local minima producing a noisy result as in Figure 3.12 (c). Although the use of the large region labels does not make a significant difference in the converged energy values, the difference is obvious if we see Figures 3.12 (d) and 3.12 (e).

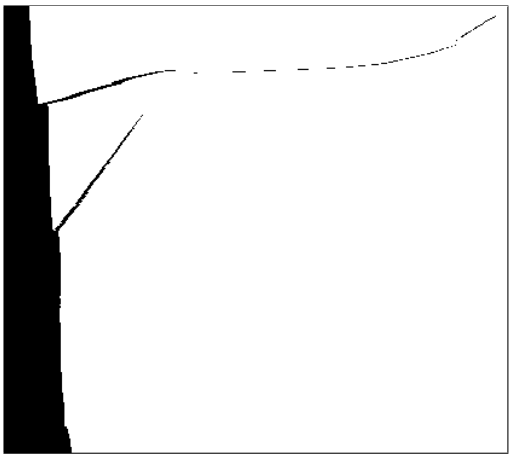
3.4.3 Comparison with PMBP

We compare our method with PMBP [Besse *et al.*, 2012] that is the closest method to ours. For a fair comparison, we use four neighbors for $\mathcal{N}$ in Equation (3.2), which is the same setting as PMBP. For a comparable smoothness weight with the default setting (eight-neighbor $\mathcal{N}$), we use $\lambda = 40$ and keep the other parameters as default. For PMBP, we use the same model as ours; the only difference from the original PMBP is the smoothness term, which does not satisfy the submodularity of Equation (2.9). PMBP also defines K candidate labels for each pixel, for which we set $K = 1$ and $K = 5$ (original paper uses $K = 5$). We show the comparison using the Cones dataset by estimating the disparity map of only the left image without the post-processing.

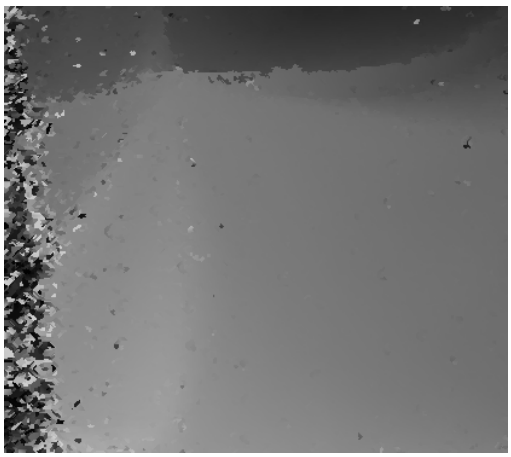
Figures 3.16 to 3.18 show the temporal transition of the energy values in a full- and zoomed-scales, and the 0.5-pixel error rates, respectively. We show the performance of our method using



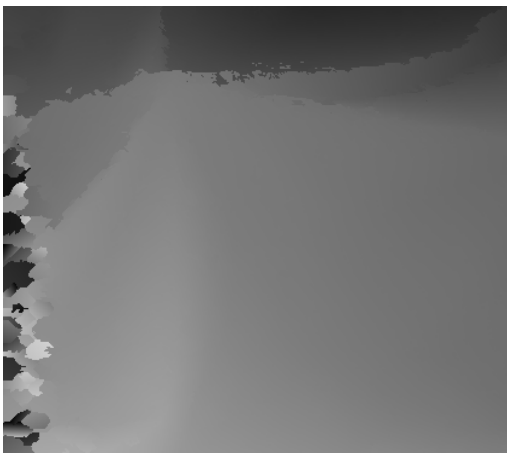
(a) left view of *Cloth1*



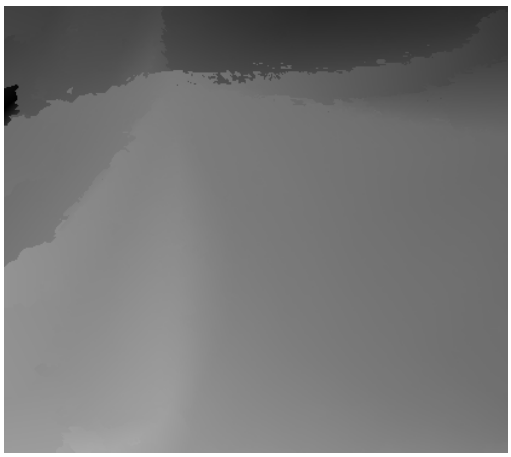
(b) visibility map of left view



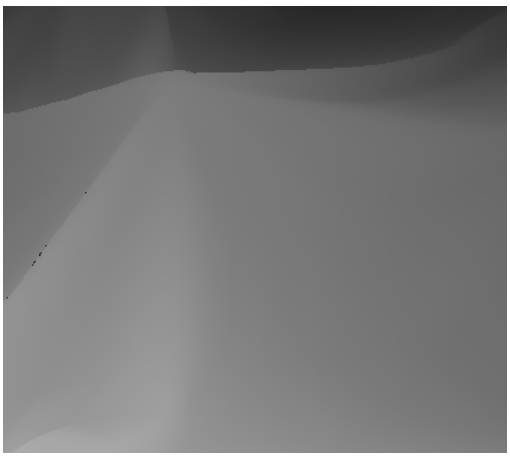
(c) only pixel labels



(d) pixel and region (5×5) labels



(e) pixel and region
(5×5 and 25×25) labels



(f) ground truth

Figure 3.12 Visual effect of region labels. (a) The left view of input images and (b) its visibility map where black pixels are invisible from the right view. (c) Using only pixel labels yields a noisy result, which is improved by (d) adding region labels. (e) Large region labels are effective for occluded regions, resulting in almost the same disparity map with (f) ground truth. These are all raw results without the post-processing.

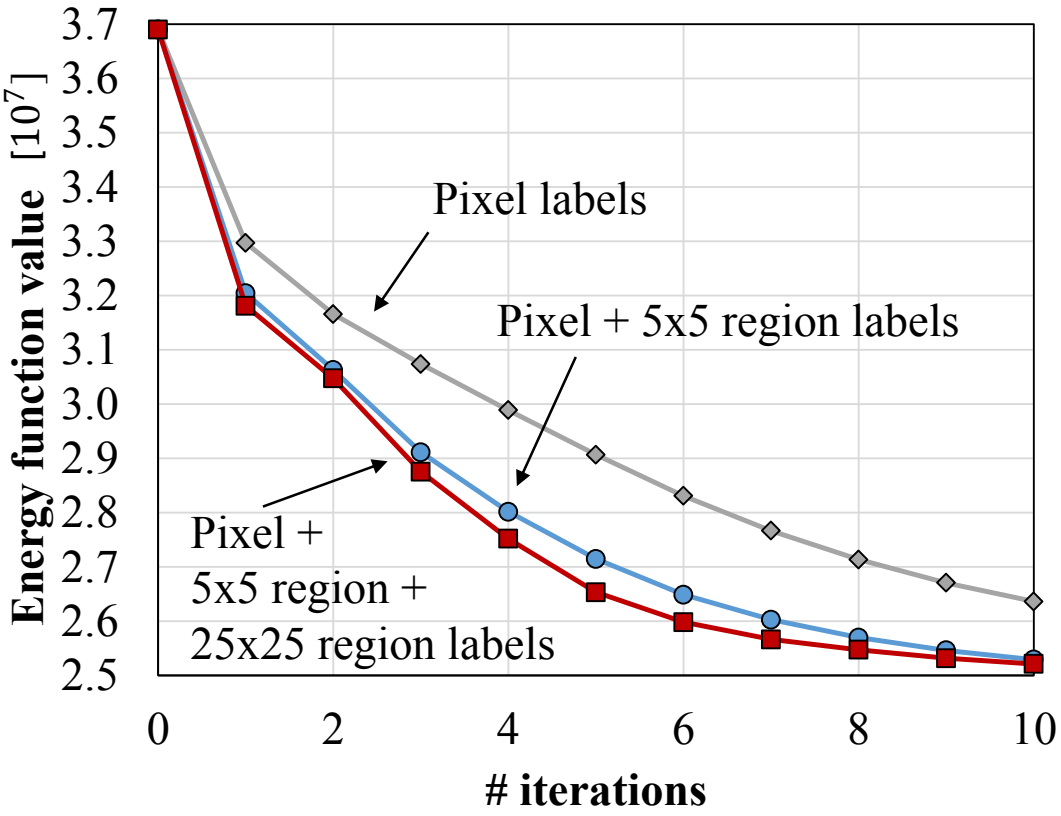


Figure 3.13 Effect of region labels in minimizing overall energies. Energy transitions of the energy function $E(f)$ w.r.t. the number of iterations are shown. Region labels play a critical role in reducing the energies.

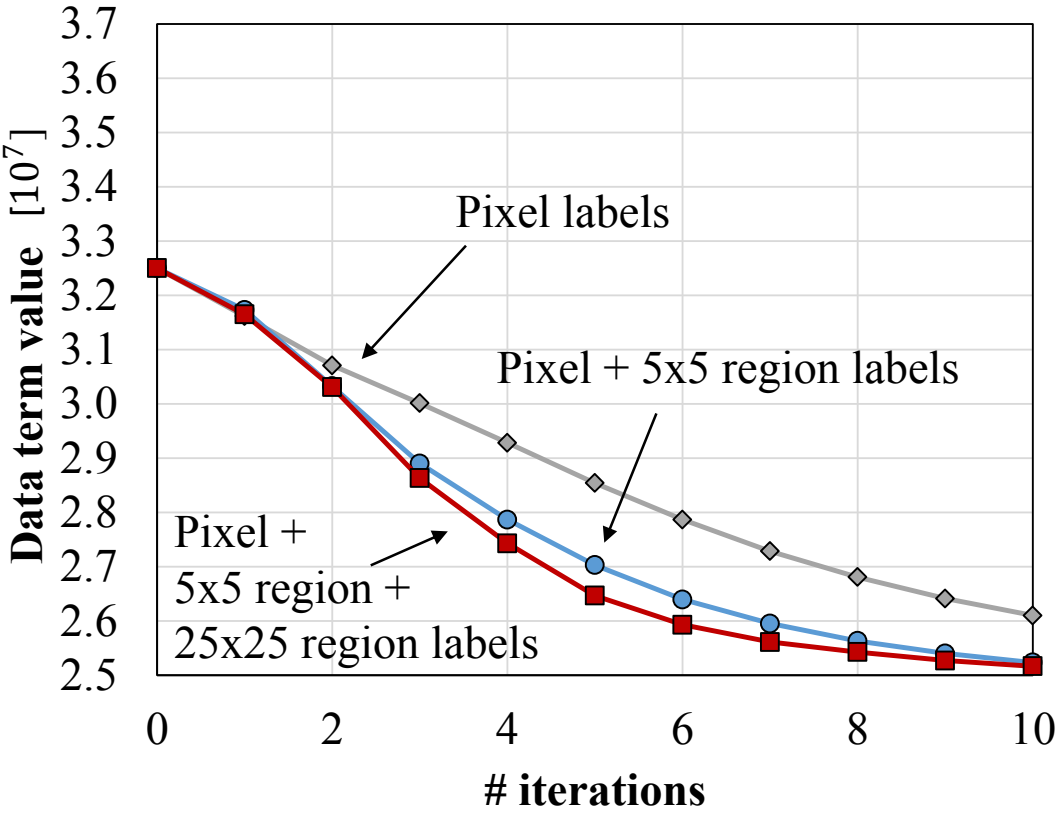


Figure 3.14 Effect of region labels in minimizing data terms. Energy transitions of the data term w.r.t. the number of iterations are shown. Region labels enable fast spatial propagations of good candidate labels as we intend.

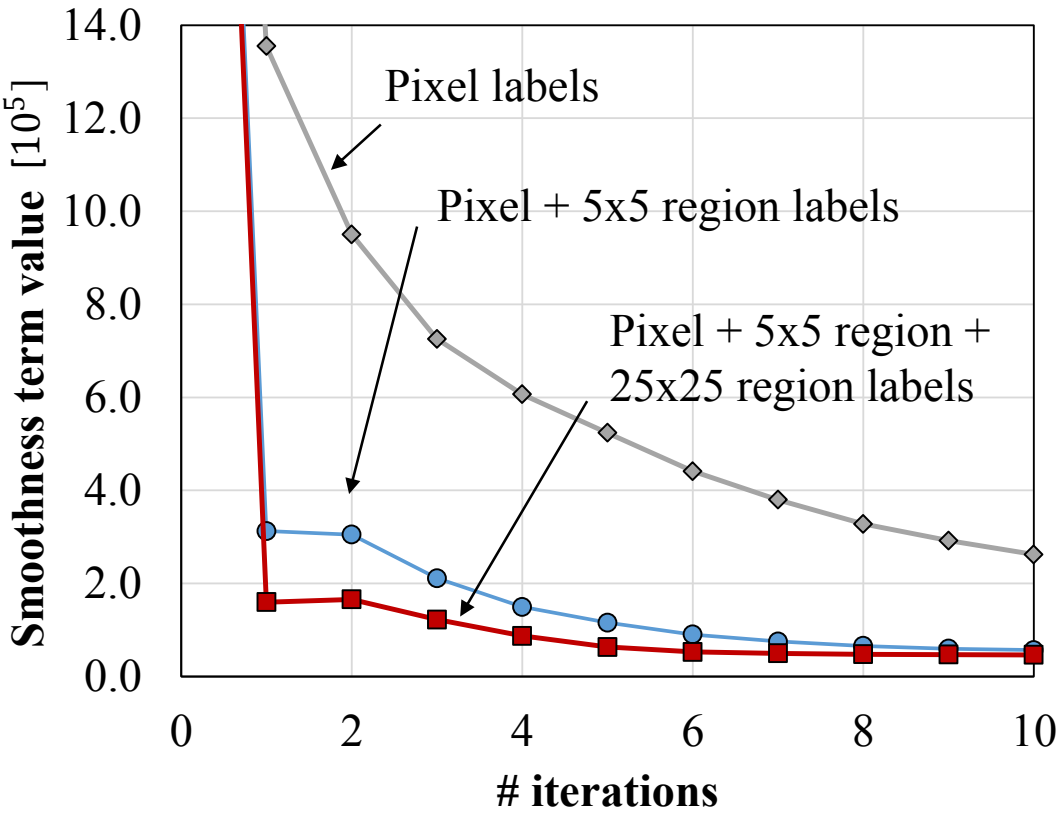


Figure 3.15 Effect of region labels in minimizing smoothness term energies. Energy transitions of the smoothness term w.r.t. the number of iterations are shown. Region labels help to find spatially smooth solutions thanks to the good properties of GC.

its GPU and CPU (1 or 4 CPU-cores) implementations. For PMBP, we also implemented the unary cost computation on GPU, but it became rather slow, possibly due to the overhead of data transfer. Efficient GPU implementations for PMBP are not available in literature[4]. Therefore, the plots show PMBP results that use a single CPU core. Figures 3.16 and 3.17 show that PMBP works much faster than our CPU implementation; however, our GPU implementation shows significantly faster convergence. Furthermore, our method reaches the better solution than that of PMBP in both energy values and error rates[5]. At around 4200[sec] of Figures 3.17 and 3.18, the solution obtained by our CPU implementation marked the lower error rate than that of PMBP in spite of its higher energy. Figure 3.19 shows the resulting disparity maps obtained by our method and PMBP with $K = 5$. Our result shows greater accuracy around the edge and occluded regions.

3.4.4 Additional Results for Outdoor Scenes

To further demonstrate the effectiveness of our method, we applied our stereo method for outdoor scene images. We use *Beijing Lion* and *Cachan Statue* datasets [Monasse, 2011] whose input image pairs are rectified and their backgrounds are removed as a pre-pcocessing. Figures 3.20 and 3.21 show the rectified input image pairs and stereo results for the two datasets. We observe major errors only around occluded regions.

3.4.5 Additional Results for Middlebury Dataset

In Figures 3.22 to 3.28, we show additional seven results for Middlebury dataset using the default settings. For a pure evaluation of the performance of our stereo method, they are all shown as raw results without the post-processing. As shown in the results, disparities are well estimated even for some occluded regions without tuning parameters.

Figures 3.29 and 3.30 show some failure cases. For *Bowling1*, we observe two difficulties that cause the large errors on the white cloth. One is that those regions are relatively texture-less. The other is that there are slight intensity differences between the input image pair. To improve the result, one can put a big weight to the gradient photo-consistency term by setting a large value to α. Also, adopting a photo-consistency measure that is robust to radiometric differences, such as normalized cross correlation, is another choice. Using the selective combination of color and gradient constraints proposed in [Xu *et al.*, 2012] may also work. For *Plastic*, we obtain significant errors simply because it has large texture-less regions.

[4] GPU-parallelization schemes of BP are not directly applicable due to PMBP's unique settings. The "jump flooding" used in the original PatchMatch [Barnes *et al.*, 2009] reports 7x speed-ups by GPU. However, because it propagates candidate labels to distant pixels, it is not applicable to PMBP that must propagate messages to *neighbors*, and is not as efficient as our 100x, either.

[5] The CPU implementation of our method also reaches almost the same energy and error rate after about 42000[sec] by a single core, 9150[sec] by four cores.

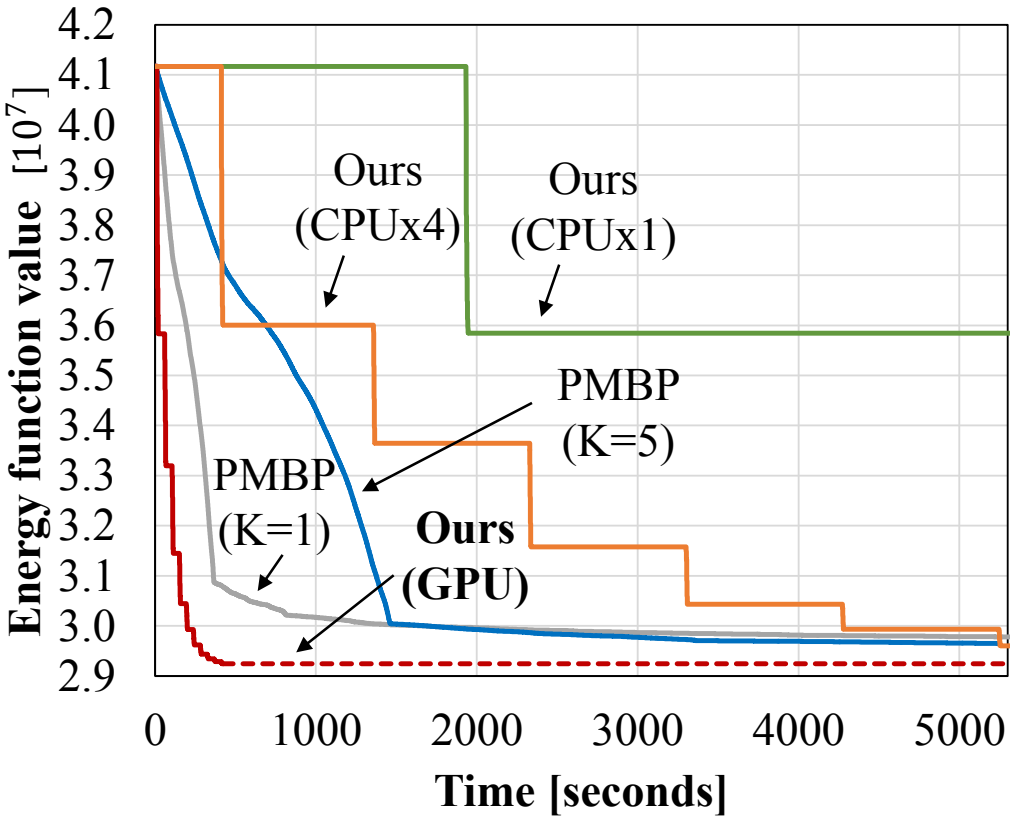


Figure 3.16 Efficiency comparison with PMBP [Besse *et al.*, 2012] in a full-scale. Accuracies are evaluated for all-regions after each iteration. PMBP is much faster than our CPU implementation; however, our GPU implementation shows significantly faster convergence. PMBP cannot be efficiently performed in parallel on GPUs.

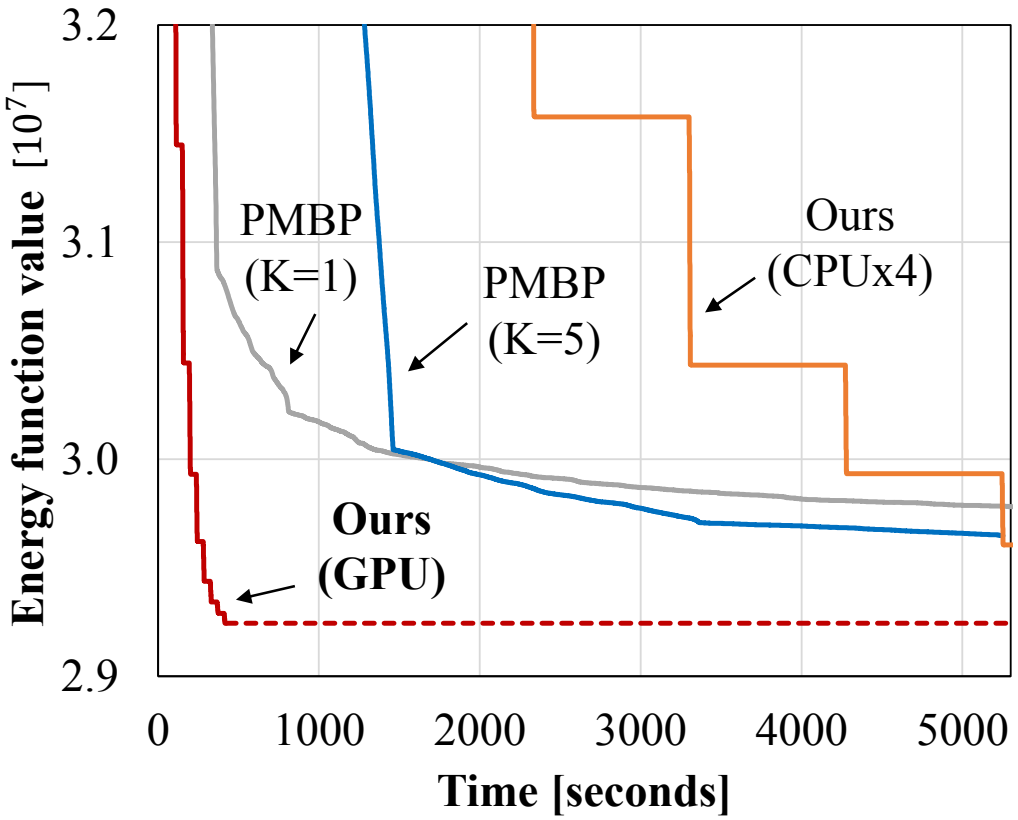


Figure 3.17 Efficiency comparison with PMBP [Besse *et al.*, 2012] in a zoomed-scale. Accuracies are evaluated for all-regions after each iteration. Our method reaches much lower energies than PMBP at convergence.

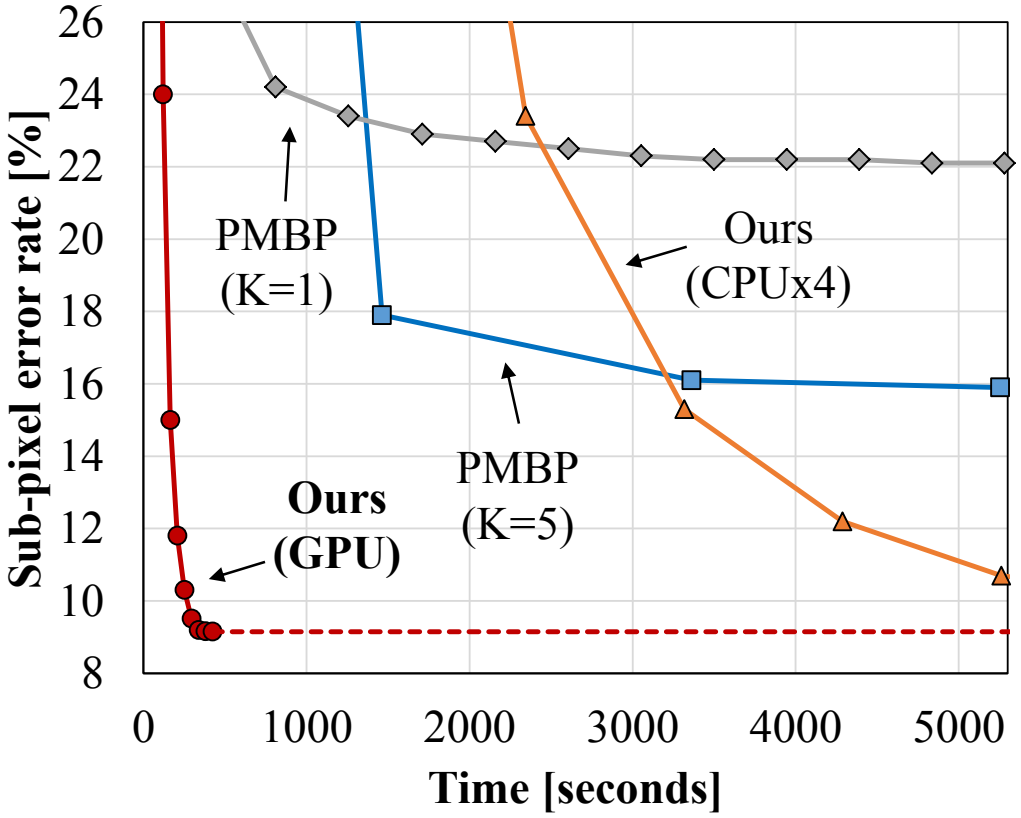
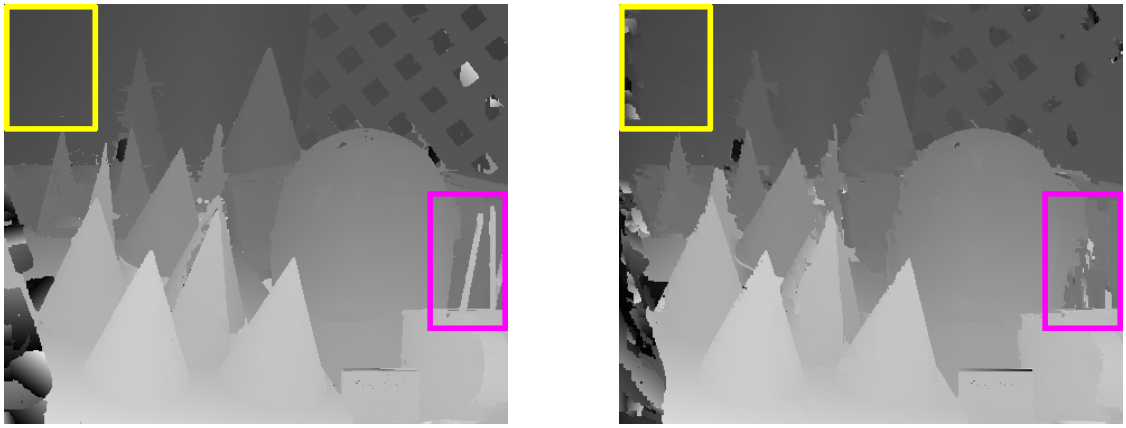
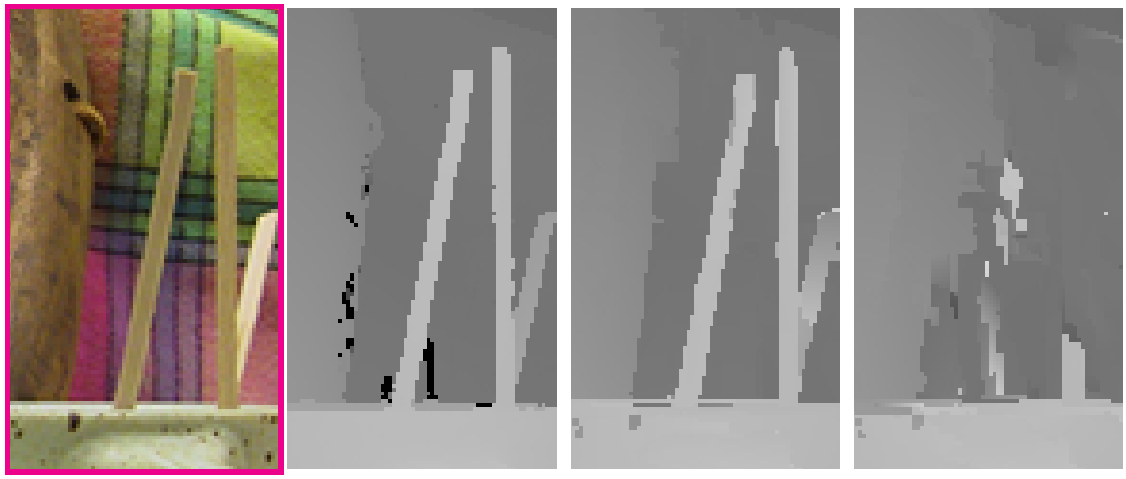


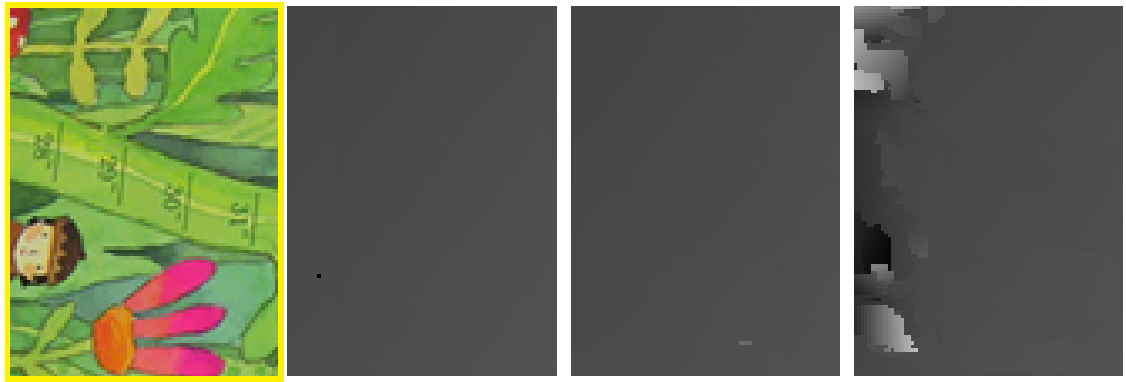
Figure 3.18 Accuracy comparison with PMBP [Besse *et al.*, 2012]. Accuracies are evaluated for all-regions after each iteration. Our method achieves greater accuracy than PMBP at convergence. Even not at convergence, *e.g.*, at around 4200[sec], the solution obtained by our CPU implementation achieves greater accuracy than PMBP in spite of its higher energy.



(a) our method (b) PMBP



edges (pink box) ground truth our method PMBP

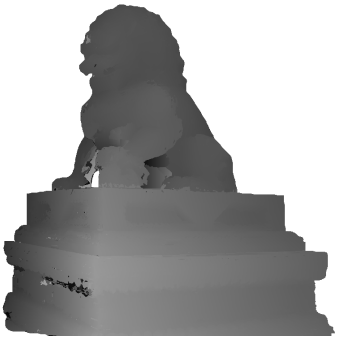


occluded (yellow box) ground truth our method PMBP

Figure 3.19 Visual comparison with PMBP [Besse *et al.*, 2012]. We show raw results of (a) our method and (b) PMBP without post-processing. Our method finds better disparities around edges (*e.g.* pink box) and occluded regions (*e.g.* yellow box).

(a) left view

(b) right view



(c) raw result

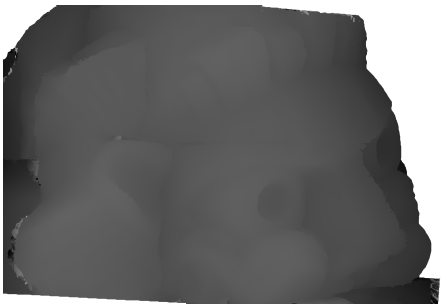
(d) after post-proccessing

(e) 3D rendered image

Figure 3.20 Results of *Beijing Lion* dataset [Monasse, 2011]. We show (a) left and (b) right views of rectified input image pairs, our results (c) before and (d) after post-processing, and (e) a 3D rendered image. We observe major errors only around occluded regions. Also, despite the great illumination differences between input image pairs, our method works robustly.

(a) left view

(b) right view



(c) raw result

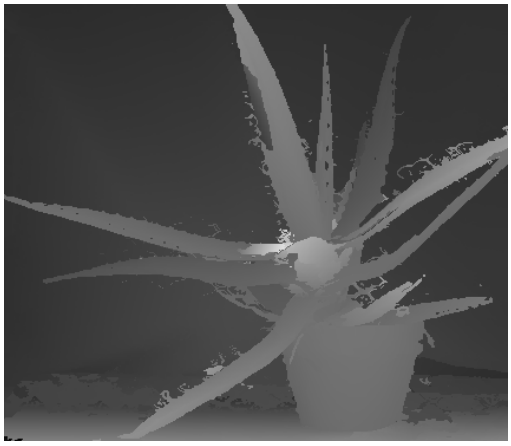
(d) after post-proccessing

(e) 3D rendered image

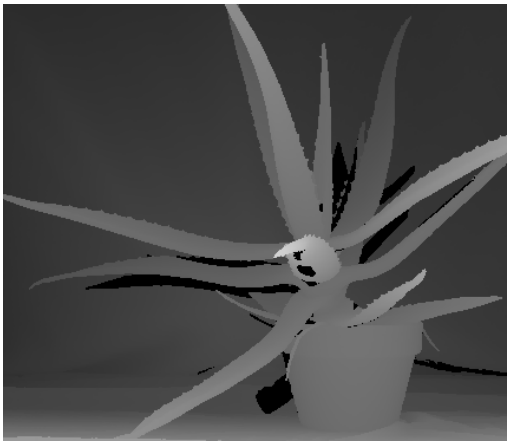
Figure 3.21 Results of *Cachan Statue* dataset [Monasse, 2011]. We show (a) left and (b) right views of rectified input image pairs, our results (c) before and (d) after post-processing, and (e) a 3D rendered image. We observe major errors only around occluded regions.

(a) left view

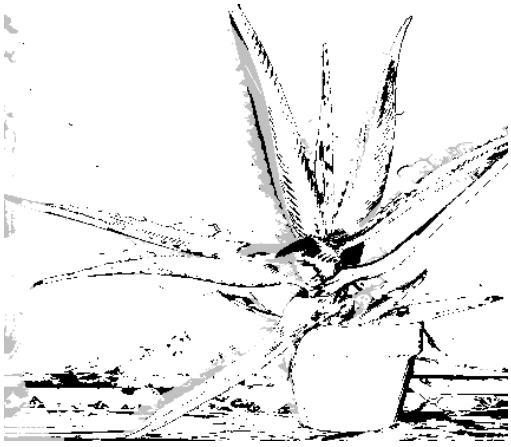
(b) right view



(c) raw result



(d) ground truth



(e) error map (11.4%)

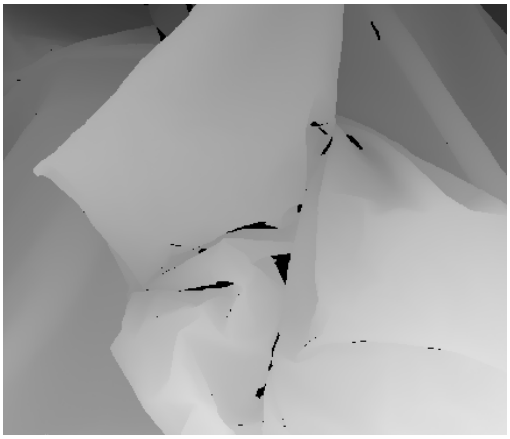
(f) visibility map

Figure 3.22 Results of *Aloe* in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result without post-processing, (d) ground truth, (e) error map with 0.5-pixel threshold, and (f) visibility map of the left view. In the error map, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.

(a) left view

(b) right view

(c) raw result

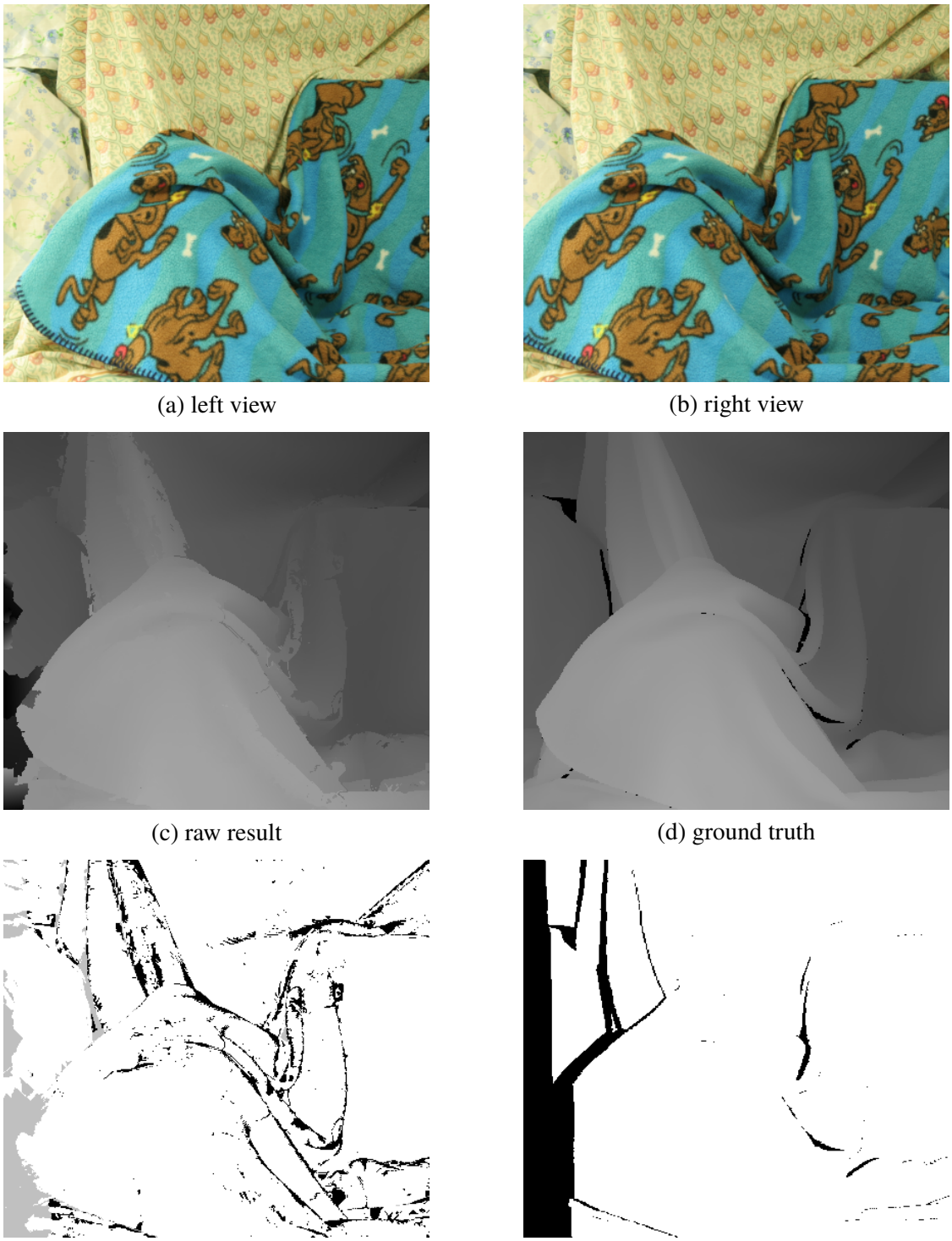


(d) ground truth

(e) error map (19.0%)

(f) visibility map

Figure 3.23 Results of *Cloth2* in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result without post-processing, (d) ground truth, (e) error map with 0.5-pixel threshold, and (f) visibility map of the left view. In the error map, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.



(a) left view (b) right view

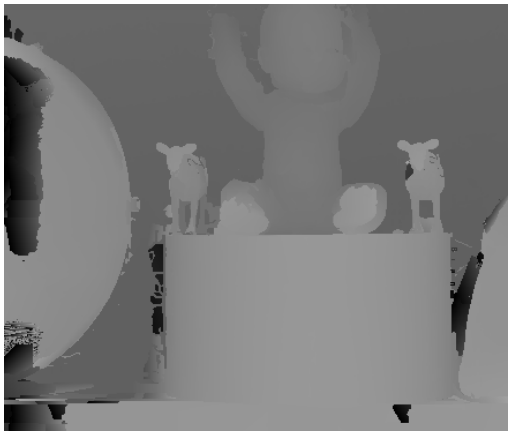
(c) raw result (d) ground truth

(e) error map (10.8%) (f) visibility map

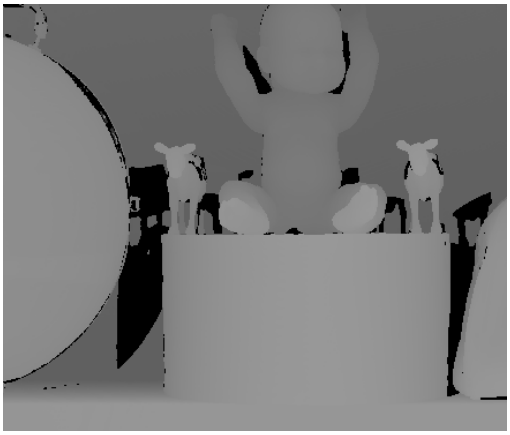
Figure 3.24 Results of *Cloth3* in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result without post-processing, (d) ground truth, (e) error map with 0.5-pixel threshold, and (f) visibility map of the left view. In the error map, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.

(a) left view

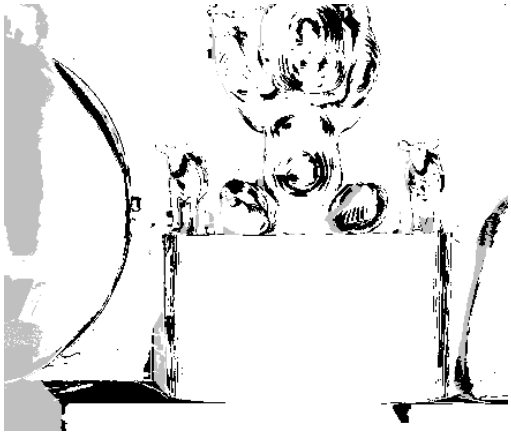
(b) right view



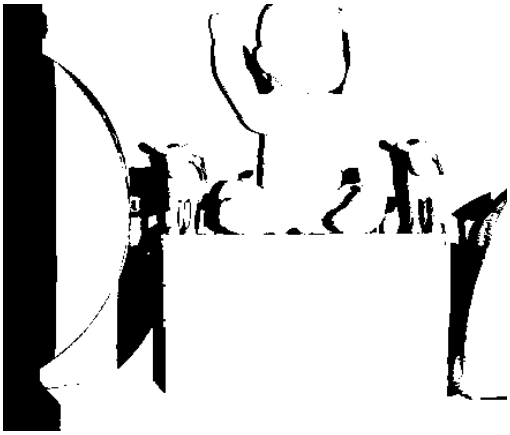
(c) raw result



(d) ground truth



(e) error map (17.1%)

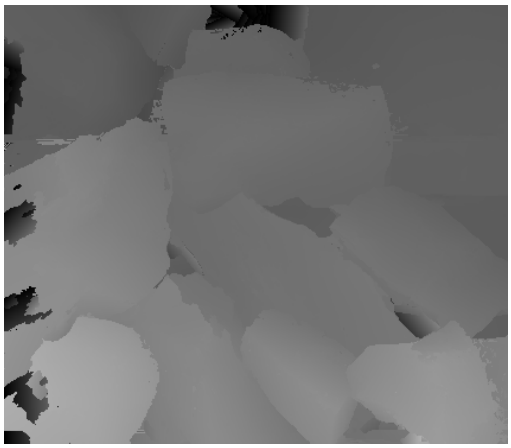


(f) visibility map

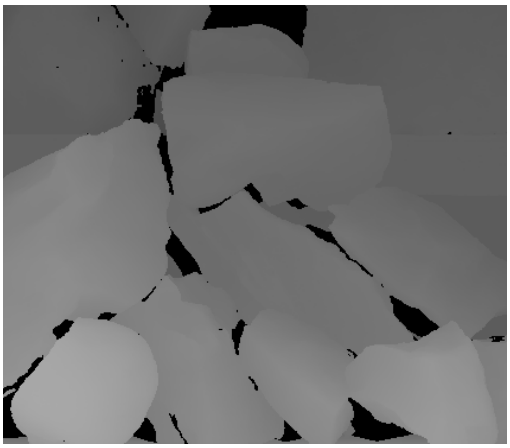
Figure 3.25 Results of *Baby3* in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result without post-processing, (d) ground truth, (e) error map with 0.5-pixel threshold, and (f) visibility map of the left view. In the error map, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.

(a) left view

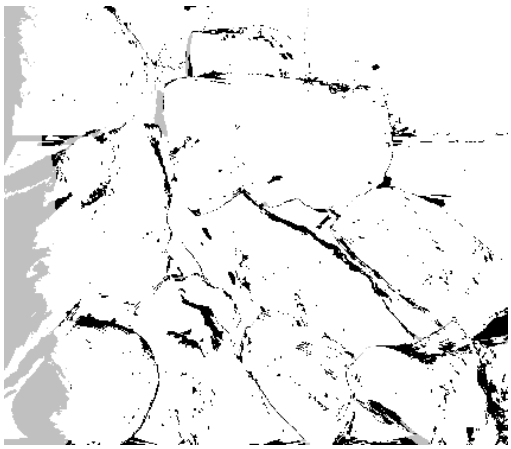
(b) right view



(c) raw result



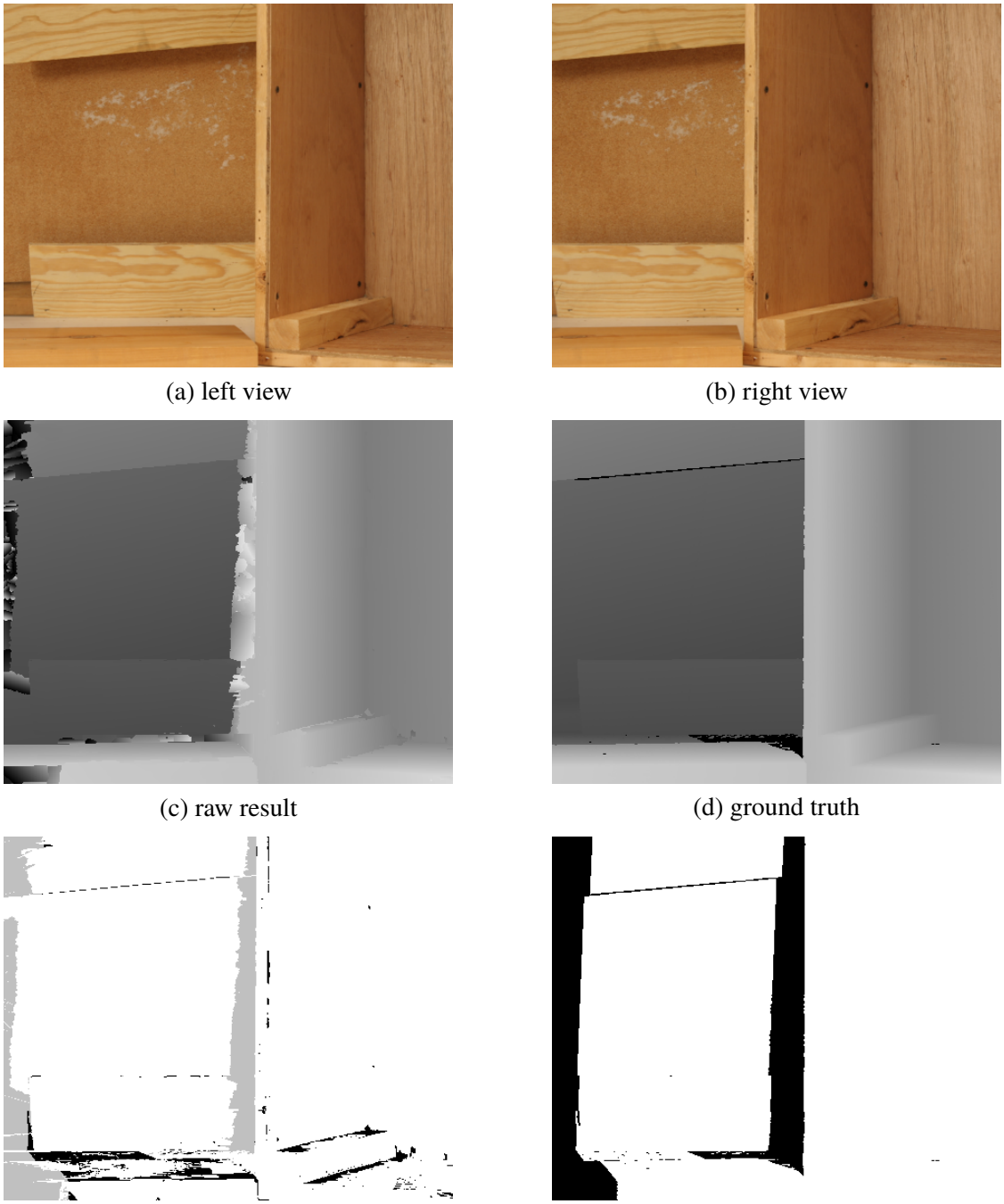
(d) ground truth



(e) error map (13.0%)

(f) visibility map

Figure 3.26 Results of *Rocks2* in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result without post-processing, (d) ground truth, (e) error map with 0.5-pixel threshold, and (f) visibility map of the left view. In the error map, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.



(a) left view (b) right view

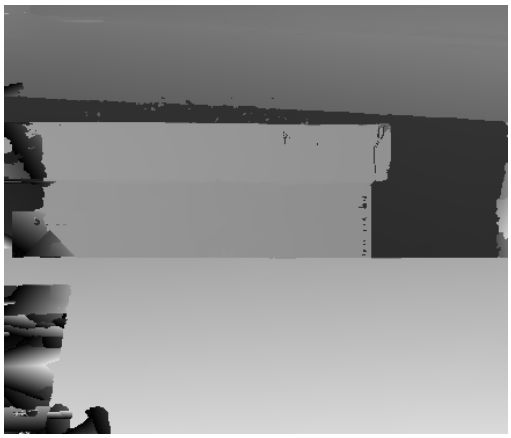
(c) raw result (d) ground truth

(e) error map (10.7%) (f) visibility map

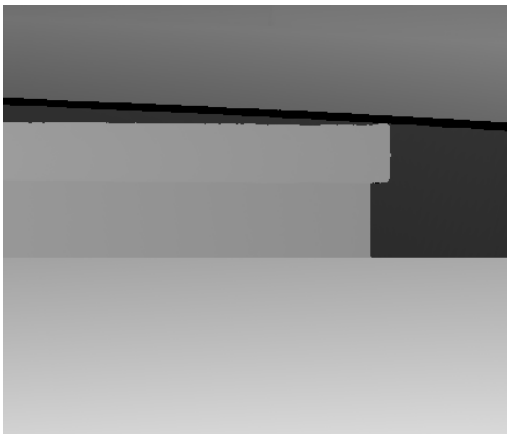
Figure 3.27 Results of *Wood1* in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result without post-processing, (d) ground truth, (e) error map with 0.5-pixel threshold, and (f) visibility map of the left view. In the error map, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.

(a) left view

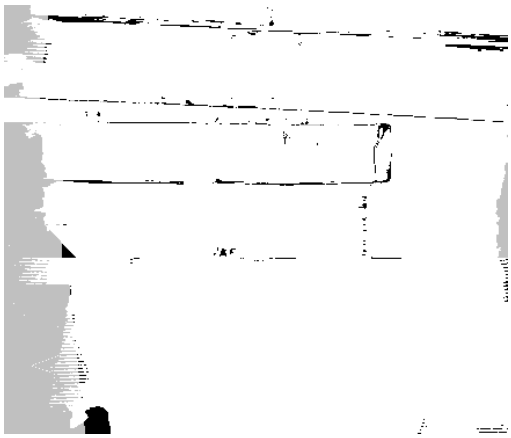
(b) right view



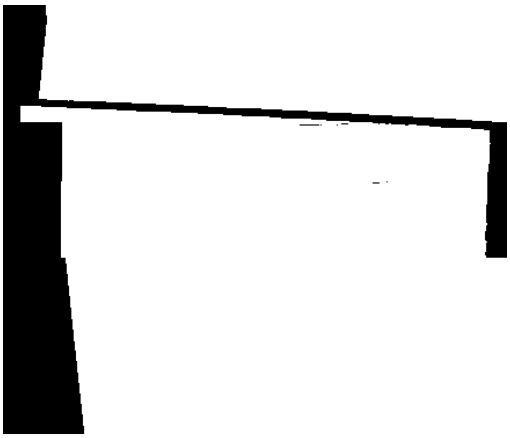
(c) raw result



(d) ground truth



(e) error map (10.7%)

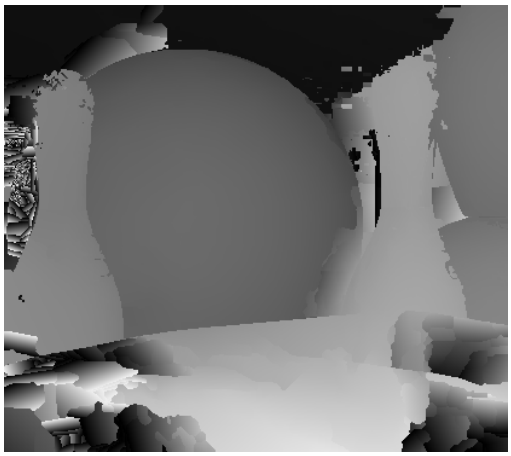


(f) visibility map

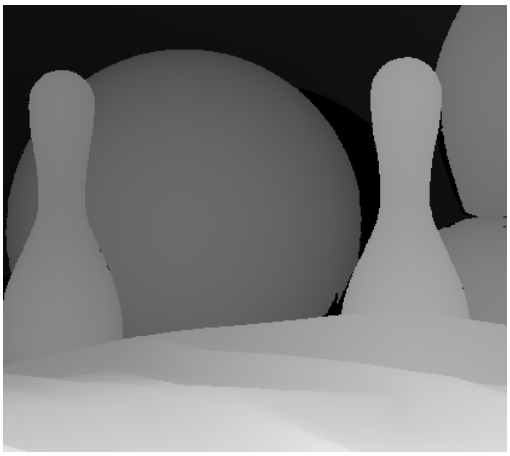
Figure 3.28 Results of *Wood2* in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result without post-processing, (d) ground truth, (e) error map with 0.5-pixel threshold, and (f) visibility map of the left view. In the error map, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.

(a) left view

(b) right view



(c) raw result

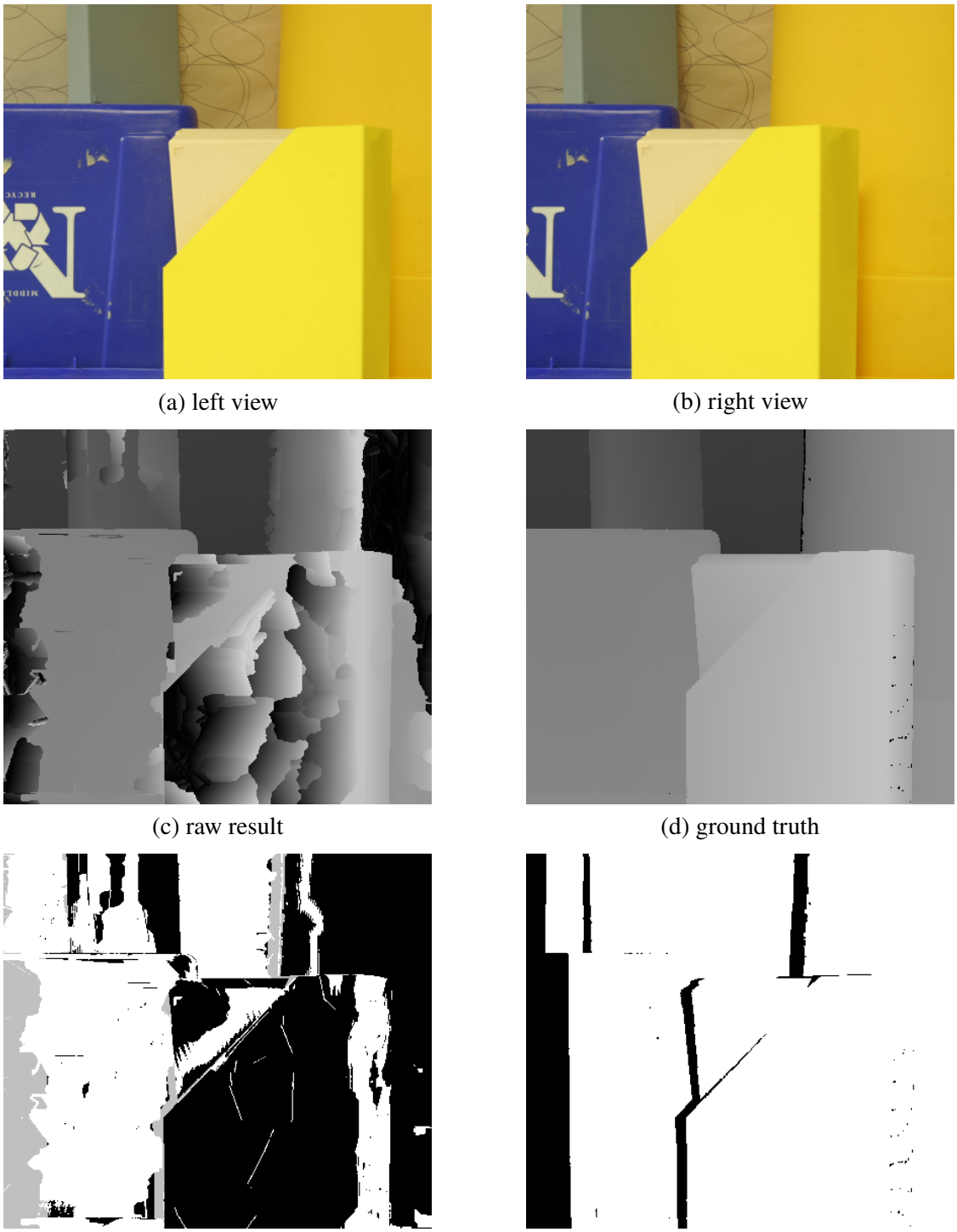


(d) ground truth

(e) error map (34.7%)

(f) visibility map

Figure 3.29 Results of *Bowling1* in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result without post-processing, (d) ground truth, (e) error map with 0.5-pixel threshold, and (f) visibility map of the left view. In the error map, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.



(a) left view (b) right view

(c) raw result (d) ground truth

(e) error map (56.0%) (f) visibility map

Figure 3.30 Results of *Plastic* in Middlebury benchmark. From top-left to right-bottom, we show (a) left and (b) right views of input images, (c) our result without post-processing, (d) ground truth, (e) error map with 0.5-pixel threshold, and (f) visibility map of the left view. In the error map, white and black pixels indicate correct and incorrect disparities, while gray indicates incorrect but occluded pixels.

3.5 Conclusions

3.5.1 Summary

In this paper, we presented an accurate and efficient stereo matching method for continuous disparity estimation. Unlike previous approaches that use fusion approaches [Lempitsky *et al.*, 2010; Olsson *et al.*, 2013; Woodford *et al.*, 2009], our method is subproblem optimal and only requires randomized initial proposals. By comparing with a recent continuous MRF stereo method, PMBP [Besse *et al.*, 2012], our method showed an advantage in efficiency and comparable or greater accuracy.

The key contribution of this work is the use of GC and locally shared labels. The intrinsic property of GC, such that it optimizes all nodes simultaneously, has an advantage and our locally shared labels are designed for taking advantage of the good property of GC as well as for enabling spatial propagation in GC-based optimization; namely, the locally shared labels derive a subproblem optimality in GC, and furthermore, the joint use of GC and the locally shared labels helps to find locally planar, smooth solutions when combined with pixelwise disparity plane formulations; spatial propagation is an essential technique for searching complex label spaces, which was originally proposed by Barnes *et al.* [2009, 2010] as a sequential algorithm but not suitable for the global inference manner of GC. Our locally shared labels enable spatial propagation in GC-based optimization for achieving efficient inference.

3.5.2 Discussions and Future Works

We discuss the current limitations and potential applications of our method, and present the future directions of this work.

Occlusions

Our current formulation does not explicitly treat occlusions but rather deals with them at post-processing. However, we believe that some occlusion handling schemes based on GC optimization [Kolmogorov and Zabih, 2001, 2002; Wei and Quan, 2005; Woodford *et al.*, 2007] can be incorporated into our framework, which may yield even greater accuracy.

Fast Cost-Volume Filtering

In our method, the computational complexity of calculating window-based matching costs depends on the size of matching windows. In the context of cost-volume filtering, fast edge-preserving filtering methods [Zhang *et al.*, 2009; He *et al.*, 2013; Lu *et al.*, 2012] have been proposed whose complexity is independent of the window size. Basically, those filtering techniques can be used when a label space is defined globally as demonstrated by Rhemann *et al.* [2011]. Recently, Lu *et al.* [2013] show that such techniques can be incorporated into PatchMatch-based inference

whose label space is defined locally and independently for each pixel. Since their key idea of region-based cost calculation is similar to our region labels, we believe that our method can be extended so as to allow the use of fast filtering techniques for further accelerating the computation.

Textureless Regions

The failure examples shown in Figures 3.30 and 3.29 indicate an obvious limitation of our method. In fact, stereo matching is not able to estimate those textureless regions in principle. To deal with this issue, the use of ground control points [Wang and Yang, 2011] is a practical choice. The ground control points are a set of sparse but reliable depth points obtained by *e.g.* depth sensors. How to use those sparse scalar depth points for helping the estimation of dense 3D planes is an open question.

4

MAP Estimation of Higher-Order MRFs for Segmentation

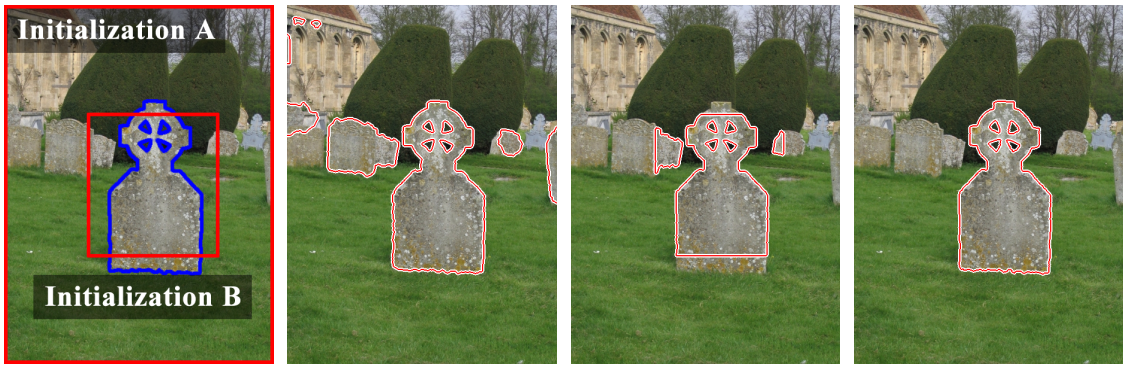
4.1 Introduction

Since the pioneer work of Boykov and Jolly [Boykov and Jolly, 2001], the use of Markov random field formulations [Geman and Geman, 1984] and graph cuts [Kolmogorov and Zabin, 2004; Boykov and Kolmogorov, 2004] has been becoming one of primary approaches to image segmentation problems [Boykov and Jolly, 2001; Rother *et al.*, 2004; Price *et al.*, 2010; Ayed *et al.*, 2013; Gorelick *et al.*, 2013, 2012; Taniai *et al.*, 2012; Pham *et al.*, 2011; Ayed *et al.*, 2010; Rother *et al.*, 2006]. In this approach, the energy function is typically formulated as

$$E(S) = R(S) + Q(S), \tag{4.1}$$

where $R(S)$ describes some appearance consistencies between resulting segments S and given information about target regions, and $Q(S)$ enforces smoothness on segment-boundaries. The form of $R(S)$ is often restricted to simple linear (*i.e.*, pixelwise unary) forms [Boykov and Jolly, 2001; Rother *et al.*, 2004; Price *et al.*, 2010] because graph cuts allow globally optimal inference only for unary and submodular pairwise forms of energies [Boykov and Kolmogorov, 2004]. However, recent studies [Ayed *et al.*, 2013; Gorelick *et al.*, 2013, 2012; Taniai *et al.*, 2012; Pham *et al.*, 2011; Ayed *et al.*, 2010; Rother *et al.*, 2006] have shown that the use of higher-order information (*i.e.*, non-linear terms such as L_1-distance of color histograms) can yield outstanding improvements over conventional pixelwise consistency measures.

Previous promising approaches try reducing energies by iteratively minimizing either first-order approximations (gradient descent approach) [Gorelick *et al.*, 2013, 2012; Rother *et al.*, 2004] or upper-bounds (bound optimization approach) [Ayed *et al.*, 2013; Taniai *et al.*, 2012; Pham *et al.*, 2011; Ayed *et al.*, 2010] of non-linear functions using graph cuts. The bound optimization approach has some advantages over the gradient descent approach [Ayed *et al.*, 2013]: it requires no parameters (*e.g.*, step-size); can be used for non-differentiable functions; never worsens the solutions during iterations. But we must in turn derive appropriate bounds for individual functions.



(a) initialization and ground truth (b) AC with initialization A (c) AC with initialization B (d) our method with initialization B

Figure 4.1 Segmentation with different initialization. AC [Ayed *et al.*, 2013] starts with all-foreground labeling (initialization A) and successively reduces foreground, often resulting in a visibly incorrect local minima. Giving a rough initial estimate (initialization B) is (c) not effective for AC [Ayed *et al.*, 2013], but (d) advantageous for our method. We use L_1-distance with 64^3 bins, and the true histogram learned from ground truth.

A notable work is *auxiliary cuts* (AC) by Ayed *et al*. [Ayed *et al.*, 2013], where they derive a general bound for a broad class of non-linear functionals. The method has shown better accuracy and faster convergence than a fast trust region method (FTR) [Gorelick *et al.*, 2013], which is another state-of-the-art based on gradient descents. One of the major limitations of AC [Ayed *et al.*, 2013] is, however, that it is formulated to successively reduce target regions, thus the resulting segments are restricted within initial segments. Such a property actually limits the applications and performance of the method. See Figure 4.1 for an illustration in histogram matching segmentation.

In this paper, we revisit a submodular-supermodular procedure (SSP) [Narasimhan and Bilmes, 2005], a general bound optimization scheme for supermodular functions using *semidifferentials* [Fujishige, 2005; Iyer *et al.*, 2013]. SSP was once studied for segmentation in [Rother *et al.*, 2006] but considered to be ineffective for its poor convergence. From our observation, however, SSP is actually useful because it allows arbitrary evolution of segments (or bidirectional optimization) over iterations. We first analyze the bound derived in [Ayed *et al.*, 2013] and reveal its close connection to SSP. Then, as an extension of both AC [Ayed *et al.*, 2013] and SSP [Narasimhan and Bilmes, 2005; Rother *et al.*, 2006], we propose a new bound optimization scheme which enables bidirectional optimization. The performance of our method is experimentally evaluated in some basic applications, and show an order of magnitude greater accuracy than AC [Ayed *et al.*, 2013]. Furthermore, we present two important applications for our method: (i) The availability of two (expand ans shrink) directions of optimization allows the use of well-known expansion and swap algorithms [Boykov and Jolly, 2001] for *multiple distribution matching problems*. We demonstrate the effectiveness of our method using L_1-distance and the Bhattacharyya measures; (ii) We show that in some situations (*e.g.*, interactive segmentation) distribution matching can be efficiently achieved in *only a single min-cut*, whereas it usually requires at least several min-cuts.

Contributions:

To summarize, we in this paper

- propose a new bound of non-linear functionals that allows arbitrary directions of convergence and outperforms the current state-of-the-art [Ayed *et al.*, 2013].
- give general understanding to the bound derived in [Ayed *et al.*, 2013] from the view of submodular function optimization.
- present, for the first time, general solutions to multiple distribution matching problems.
- show that, in some scenarios, nearly optimal solutions can be obtained in as few as a single min-cut.

The code will be publicly available.

4.2 Considered Problems and Prior Art

The scope of the problems where our method is applicable is the same or even wider than AC [Ayed *et al.*, 2013]. By following the problem statements of [Ayed *et al.*, 2013], we present the forms of the considered problems in this paper, and compare the application ranges of our method and AC [Ayed *et al.*, 2013].

4.2.1 Linear Functions

Let $I_p = I(p) : \mathbb{R}^2 \rightarrow Z$ be an image function, where $Z \subset \mathbb{R}^d$ is the space of pixel features such as RGB intensities. The image I_p is defined on the 2D image coordinates $p_i = p(i) : \Omega \rightarrow \mathbb{Z}^2$, where Ω is the image domain. The objective here is to seek segments $S \subseteq \Omega$ such that S minimizes $E(S)$ of Equation (4.1). When the appearance term $R(S)$ is a linear product of a function $h : \Omega \rightarrow \mathbb{R}$

$$R(S) = \langle h, S \rangle = \sum_{i \in S} h(i) \tag{4.2}$$

and $Q(S)$ is the sum of pairwise submodular functions, then $E(S)$ can be globally minimized via graph cuts [Boykov and Kolmogorov, 2004; Kolmogorov and Zabin, 2004] in polynomial times. A typical example is the BJ model [Boykov and Jolly, 2001], whose unary costs h can be expressed by

$$h(i) = -\log \Pr(I_{p_i}|\mathcal{F}) + \log \Pr(I_{p_i}|\mathcal{B}) \tag{4.3}$$

where $\Pr(\cdot|\mathcal{F})$, $\Pr(\cdot|\mathcal{B})$ are priori known probability distributions of pixel features inside and outside the target regions, respectively.

4.2.2 Type-I: Non-linear Functions of Linear Terms

Similarly to [Ayed *et al.*, 2013], we mainly focus on the following form of non-linear functions:

$$R(S) = \sum_{z \in Z} f_z(\langle g_z, S \rangle) \tag{4.4}$$

where $g_z : \Omega \to \mathbb{R}^+$ is a family of non-negative scalar functions defined over the image domain, and $f_z : C \subset \mathbb{R} \to \mathbb{R}$ is an arbitrary convex function defined over a convex domain C. For this type of functions, our method allows bidirectional optimization (*i.e.*, segments S can arbitrarily evolve over iterations), which is an important advantage over AC [Ayed *et al.*, 2013] that can only successively reduce segments (*i.e.*, $S^0 \supseteq S^1 \supseteq S^2 ...$). Also, AC [Ayed *et al.*, 2013] assumes only positive convex functions $f_z : C \subset \mathbb{R} \to \mathbb{R}^+$. Examples of this type of functions include but not limited to the following some useful constrains.

The L_p-Distance Constrains between Histograms

Given a histogram $\{h_z,\ z \in Z\}$ of pixel features within the target regions, the L_p-distance constrains between the given histogram and the histogram within segments S can be derived from Equation (4.4) by substituting $g_z(i) = \delta(z - I_{p_i})$ and $f_z(x) = |h_z - x|^p, p \geq 1$ as

$$\left| h_z - \sum_{i \in S} \delta(z - I_{p_i}) \right|^p \tag{4.5}$$

where the summation $\sum_{i \in S} \delta(z - I_{p_i})$ counts the number of pixels in S that fall into bin z.

The Area-Size Constraints

Given a area-size v_1 of the target region, the size constraints between v_1 and $|S|$ are derived from Equation (4.4) by substituting $g_z = 1, z \in \{1\}$ and $f_z(x) = |v_z - x|^p, p \geq 1, z \in \{1\}$ as

$$\left| v_1 - \sum_{i \in S} 1 \right|^p . \tag{4.6}$$

4.2.3 Type-II: Non-linear Functions of the Ratio of Linear Terms

Like [Ayed *et al.*, 2013], our method can be also used for the following non-linear functions:

$$R(S) = \sum_{z \in Z} f_z \left(\frac{\langle g_z, S \rangle}{\langle w_z, S \rangle} \right) \tag{4.7}$$

where $g_z, w_s : \Omega \to \mathbb{R}^+$ are arbitrary non-negative functions defined over the image domain, and $f_z : C \subset \mathbb{R} \to \mathbb{R}$ is a convex, monotonically decreasing function defined over a convex domain

C. For this type of functions, however, our method does not allow bidirectional optimization. Therefore, our method proceeds by iteratively reducing segments S just like AC [Ayed *et al.*, 2013]. Examples of this type of functions include the following useful constrains.

Probability Product Kernels

Given a probability distribution $\Pr(z|\mathcal{F})$ of pixel features within the target region, probability product kernels [Jebara *et al.*, 2004] can be derived from Equation (4.7) by substituting $g_z = k_z$ some kernel function, $w_z = 1$, $f_z(x) = [x\Pr(z|\mathcal{F})]^\rho$, $\rho \in]0, 1]$ as

$$-\sum_{z\in Z}\left(\frac{\langle k_z, S\rangle}{\langle 1, S\rangle}\right)^\rho \Pr(z|\mathcal{F})^\rho \tag{4.8}$$

where the kernel function is *e.g.* $k_z(i) = \delta(z - I_{p_i})$. When $\rho = 0.5$, the function becomes the Bhattacharyya coefficient [Bhattacharyya, 1943; Aherne *et al.*, 1998], which has been used for segmentation in [Ayed *et al.*, 2010; Taniai *et al.*, 2012; Pham *et al.*, 2011].

Kullback-Leibler Divergence

Similarly, the KL divergence can be expressed as

$$\sum_{z\in Z}\Pr(z|\mathcal{F})\log\left(\frac{\Pr(z|\mathcal{F})}{\frac{\langle k_z,S\rangle}{\langle 1,S\rangle}+\epsilon}\right) = \underbrace{\sum_{z\in Z}\Pr(z|\mathcal{F})\log\left(\Pr(z|\mathcal{F})\right)}_{constant} - \underbrace{\sum_{z\in Z}\Pr(z|\mathcal{F})\log\left(\frac{\langle k_z, S\rangle}{\langle 1, S\rangle}+\epsilon\right)}_{variable}, \tag{4.9}$$

where $\epsilon > 0$ is a small constant for avoiding division by 0. So the variable part of the KL divergence constrains is derived from Equation (4.7) by substituting $g_z = k_z$ some kernel function, $w_z = 1$, $f_z(x) = -\Pr(z|\mathcal{F})\log(x + \epsilon)$.

4.2.4 Type-III: Non-Linear Functions of Linear Terms for Multi-Models

One of the main contributions of this paper is the application to multi-model segmentation, where we seek K-division ($K \geq 2$) of segments $D_K = \{S_1, S_2, ..., S_K\}$ (*i.e.*, $S_i \cap S_j = \emptyset$ if $i \neq j$, and $S_1 \cup S_2 \cup ... \cup S_K = \Omega$) by minimizing

$$E(D_K) = \sum_{i=1}^{K} R_i(S_i) + Q(D_K) \tag{4.10}$$

where $R_i(S_i)$ is given as the Type-I form of Equation (4.4), and $Q(D_K)$ is the smoothness term. Therefore, a typical example of this type of functions is multi-model L_p-distance. Additionally,

we show that the Bhattacharyya coefficient (*i.e.*, Equation (4.8) with $\rho = 0.5$) can be also used here by extending the formulation proposed in [Taniai *et al.*, 2012], where the authors particularly addressed the case of the Bhattacharyya coefficient with $K = 2$.

4.3 Review of Submodular-Supermodular Procedures

We review SSP [Narasimhan and Bilmes, 2005] as an energy minimization framework for supermodular functions. Let $R(S) : 2^{\Omega} \rightarrow \mathbb{R}$ be a discrete function. $R(S)$ is submodular and supermodular, if it satisfies the following inequalities, respectively, for any $X, Y \subseteq \Omega$:

$$R(X) + R(Y) \geq R(X \cap Y) + R(X \cup Y) \tag{4.11}$$

$$R(X) + R(Y) \leq R(X \cap Y) + R(X \cup Y) \tag{4.12}$$

Therefore, if $R(S)$ is supermodular, then $-R(S)$ is submodular. While any submodular functions can be minimized in polynomial times [Schrijver, 2000], the minimization of supermodular functions (*i.e.*, the maximization of submodular functions) is NP-hard.

4.3.1 Bound Optimization

SSP [Narasimhan and Bilmes, 2005] can be seen as a bound optimization framework for supermodular functions. In bound optimization approaches, a tight upper bound function $\hat{E}(S|S^t)$ given an auxiliary variable S^t is derived for $E(S)$, *i.e.*,

$$E(S) \leq \hat{E}(S|S^t) \quad \text{and} \quad E(S^t) = \hat{E}(S^t|S^t). \tag{4.13}$$

Then, in the following iterative minimization procedure

$$S^{t+1} = \arg\min \hat{E}(S|S^t), \tag{4.14}$$

the energy does not go up: $E(S^t) \geq E(S^{t+1})$ because $E(S^t) = \hat{E}(S^t|S^t) \geq \hat{E}(S^{t+1}|S^t) \geq E(S^{t+1})$.

4.3.2 Semidifferentials

SSP uses semidifferentials [Fujishige, 2005; Iyer *et al.*, 2013] for obtaining the tight upper bounds of supermodular functions. Given a supermodular function $R(S)$ and $X \subseteq \Omega$, if a modular (or linear) function $H(S) := \langle h, S \rangle + R(\emptyset)$ satisfies

$$H(S) - H(X) \geq R(S) - R(X) \tag{4.15}$$

for any $S \subseteq \Omega$, then $H(S)$ is called a *semigradient* of R at X. We denote $\partial R(X)$ the set of all the semigradients of R at X, and $\partial R(X)$ is called the *semidifferential* of R at X. Notice that if $H(X) = R(X)$ holds, then the semigradient $H(S)$ gives a tight upper bound to the supermodular function $R(S)$. Such extreme points of $\partial R(X)$ may be obtained using the following theorem.

Theorem (Theorem 6.11 in [Fujishige, 2005]): *For any $X \subseteq \Omega$, a modular function $H(S) := \langle h, S\rangle + R(\emptyset)$ is an extreme point of $\partial R(X)$ if and only if there exists a maximal chain*

$$C : \emptyset = S_0 \subset S_1 \subset \cdots \subset S_n = \Omega, \tag{4.16}$$

with $S_j = X$ for some j, such that

$$H(S_i) - H(S_{i-1}) = R(S_i) - R(S_{i-1}) \quad (i = 1, 2, \cdots, n). \tag{4.17}$$

SSP [Narasimhan and Bilmes, 2005] makes a semigradient $H^\sigma(S|S^t)$ of $R(S)$ at S^t using a greedy algorithm as follows. Let σ be a permutation of Ω that assigns the elements in S^t to the first $|S^t|$ positions, *i.e.*, $\sigma(i) \in S^t$ if and only if $i \leq |S^t|$. A maximal chain C^σ is then defined as $S_0^\sigma = \emptyset$ and $S_i^\sigma = \{\sigma(1), \sigma(2), ..., \sigma(i)\}$, so $S_{|S^t|}^\sigma = S^t$. See Figure 4.2 (a) for an illustration. Using this chain C^σ, a semigradient $H^\sigma(S|S^t)$ is immediately obtained as

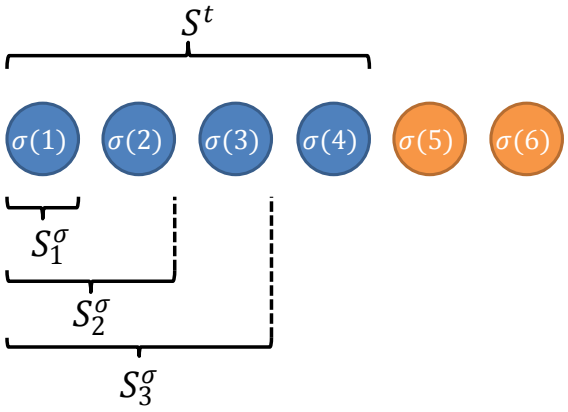
$$H^\sigma(S|S^t) = \langle h^\sigma, S\rangle + R(\emptyset) \tag{4.18}$$

where each unary cost is given by

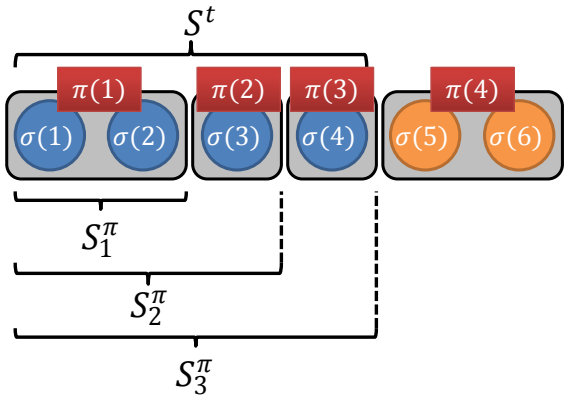
$$h^\sigma(\sigma(i)) = R(S_i^\sigma) - R(S_{i-1}^\sigma). \tag{4.19}$$

Figure 4.3 (a) illustrates how $h^\sigma(\sigma(i))$ are computed. It can be easily shown that $H^\sigma(S_i^\sigma|S^t) = R(S_i^\sigma)$ holds for any i, and a semigradient $H^\sigma(S_i^\sigma|S^t)$ is thus a tight upper bound to $R(S)$ satisfying Equation (4.13). It also means that if the optimal solution S^* happens to be among the sets $\{S_i^\sigma\}$, then the minimization of $H^\sigma(S|S^t)$ will find the optimal solution S^*. Also, since $R(S)$ is supermodular, $h^\sigma(\sigma(i)) \leq h^\sigma(\sigma(i+1))$ holds for any i. Therefore, the best practice for estimating permutations σ is to align the elements of Ω so that $L(\sigma(i)) \geq L(\sigma(i+1))$ holds with $L(i)$ some likelihood of i being inside S^*.

SSP was first used for segmentation in [Rother *et al.*, 2006], where the L_1-distance consistencies between histograms (*i.e.*, Equation (4.5) with $p = 1$) were enforced by iteratively minimizing the bounds $\hat{E}(S|S^t) = H^\sigma(S|S^t) + Q(S)$ for $E(S)$. To estimate a permutation σ, they used a signed distance map $D(p_i|S^t)$ from the border of the current segments S^t; so the pixel i that is most inside the segments S^t comes at the first position: $\sigma(1) = i$. Their results showed that, however, the schemes was ineffective and only used for making initial segments for their proposed method,



(a) SSP



(b) our method

Figure 4.2 Illustration of the chain and permutation. (a) SSP [Narasimhan and Bilmes, 2005] is a special case of (b) our method. The use of our grouped permutation π makes piecewise-mean-approximation bounds. See also Figure 4.3.

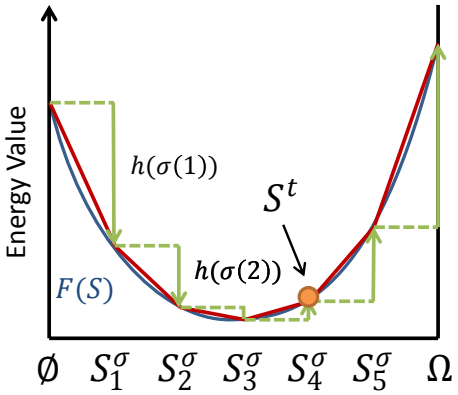
trust region graph cuts, based on gradient descents.

4.3.3 Relationship with Auxiliary Cuts

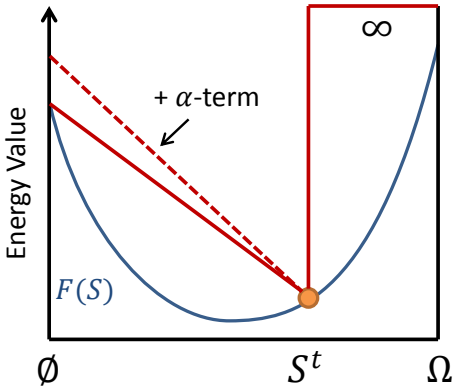
The authors of [Ayed *et al.*, 2013] studied the Type-I of non-linear functions in Equation (4.4), which can be re-written as a discrete form:

$$R(S) = \sum_{z \in Z} f_z(\langle g_z, S \rangle) = \sum_{z \in Z} F_z(S) \tag{4.20}$$

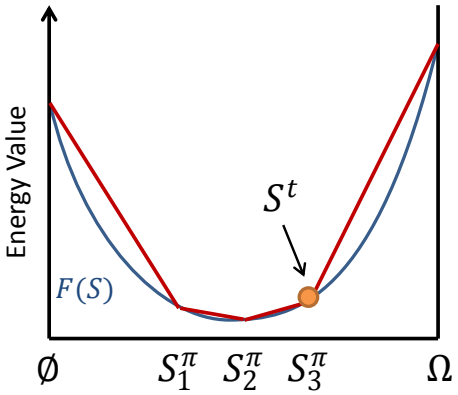
with $F_z(S) = f_z(\langle g_z, S \rangle) : 2^\Omega \rightarrow \mathbb{R}$ a supermodular function. Using the Jensen's inequality and assuming $S \subseteq S^t$ they derived a general bound $A_\alpha(S|S^t)$ for this $R(S)$, which can be equivalently



(a) bound in SSP



(b) bound in AC



(c) our generalized bound

Figure 4.3 Illustration of upper bounds for supermodular functions. The bounds in (a) SSP [Narasimhan and Bilmes, 2005], (b) AC [Ayed *et al.*, 2013], and (c) our method are shown, where the supermodular function $F(S)$ and its bounds are "intuitively" visualized by blue and red lines, respectively. The green arrows in (a) show the unary costs $h(\sigma(i))$ of semidifferentials.

expressed by the following form:

$$A_\alpha(S|S^t) = B_{shrink}(S|S^t) + \alpha R(S^t)\left(1 - \frac{\langle 1, S\rangle}{\langle 1, S^t\rangle}\right) \geq R(S) \tag{4.21}$$

where $\alpha \geq 0$ and $B_{shrink}(S|S^t)$ is given as

$$\begin{aligned} B_{shrink}(S|S^t) &= R(S^t) + \left\langle \sum_{z\in Z} \frac{F_z(\emptyset) - F_z(S^t)}{\langle g_z, S^t\rangle} g_z, S^t \setminus S \right\rangle \\ &= \sum_{z\in Z}\left(F_z(S^t) + \frac{F_z(\emptyset) - F_z(S^t)}{\langle g_z, S^t\rangle}\langle g_z, S^t \setminus S\rangle\right) \\ &= \sum_{z\in Z}\left(F_z(\emptyset) + \frac{F_z(S^t) - F_z(\emptyset)}{\langle g_z, S^t\rangle}\langle g_z, S\rangle\right) \geq R(S). \end{aligned} \tag{4.22}$$

Therefore, it can be seen that $B_{shrink}(S|S^t)$ is the sum of the linear approximations of $F_z(S)$ drawn from $F_z(\emptyset)$ to $F_z(S^t)$. Figure 4.3 (b) visualizes this linear approximation function shown as the solid red line. Note that the infinite bound in Figure 4.3 (b) reflects the fact $S \subseteq S^t$ and the dashed line depicts the effect of the α-term in Equation (4.21). Although the function of this α-term was not mentioned in [Ayed *et al.*, 2013], it is used for preventing the optimization process from over-shrinking the segments S, because the method [Ayed *et al.*, 2013] can only successively reduce the segments S and thus cannot recovery from over-shrinking during iterations.

Furthermore, we can derive another bound for $R(S)$ similar to $B_{shrink}(S|S^t)$ by applying the same derivation to $\bar{R}(S) := R(\Omega \setminus S)$ and substituting $S \leftarrow \Omega \setminus S$, $S^t \leftarrow \Omega \setminus S^t$, which results after some simple calculations in

$$B_{expand}(S|S^t) = \sum_{z\in Z}\left(F_z(S^t) + \frac{F_z(\Omega) - F_z(S^t)}{\langle g_z, \Omega \setminus S^t\rangle}\langle g_z, S \setminus S^t\rangle\right) \geq R(S). \tag{4.23}$$

Here, the auxiliary variable S^t is assumed to be $S^t \subseteq S$, so the iterative minimization of this bound successively *expands* the segments S. Similarly to $B_{shrink}(S|S^t)$, this bound can be seen as the sum of the linear approximations of $F_z(S)$ drawn from $F_z(S^t)$ to $F_z(\Omega)$.

Now we can see an interesting relationship between the semidifferentials used in SSP [Narasimhan and Bilmes, 2005; Rother *et al.*, 2006] and the bound in [Ayed *et al.*, 2013]. Namely, the bounds $B_{shrink}(S|S^t)$ and $B_{expand}(S|S^t)$ can be obtained as the mean approximations of semidifferentials. To show this, let us denote the semidifferential of $F_z(S)$ by $H_z^\sigma(S|S^t) = \langle h_z^\sigma, S\rangle + F_z(\emptyset)$ where $h_z^\sigma(\sigma(i)) = F_z(S_i^\sigma) - F_z(S_{i-1}^\sigma)$. We derive the mean approximation of $H_z^\sigma(S|S^t)$ by

dividing the domain Ω into S^t and $\bar{S}^t := \Omega \setminus S^t$ as follows:

$$
\begin{aligned}
H_{mean}(S|S^t) = & \sum_z \underbrace{\left[H_z^\sigma(\emptyset|S^t) + \frac{H_z^\sigma(S^t|S^t) - H_z^\sigma(\emptyset|S^t)}{\langle g_z, S^t \rangle} \langle g_z, S \cap S^t \rangle \right]}_{\text{mean approximation of } H_z^\sigma(S|S^t) \text{ from } S_0^\sigma = \emptyset \text{ to } S_{|S^t|}^\sigma = S^t} \\
& + \sum_z \underbrace{\left[H_z^\sigma(S^t|S^t) + \frac{H_z^\sigma(\Omega|S^t) - H_z^\sigma(S^t|S^t)}{\langle g_z, \Omega \setminus S^t \rangle} \langle g_z, S \cap \bar{S}^t \rangle \right]}_{\text{mean approximation of } H_z^\sigma(S|S^t) \text{ from } S_{|S^t|}^\sigma = S^t \text{ to } S_{|\Omega|}^\sigma = \Omega} \\
= & \; B_{shrink}(S \cap S^t|S^t) + B_{expand}(S \cap \bar{S}^t|S^t). \qquad (4.24)
\end{aligned}
$$

Notably, the bounds of [Ayed *et al.*, 2013] are obtained here by a completely different way than using the Jensen's inequality, without assuming $S \subseteq S^t$ nor $S^t \subseteq S$. The derived function $H_{mean}(S|S^t)$ satisfies the tight bound conditions of Equation (4.13) (a proof is given in the next section), and allows bidirectional optimization. However, when we use this bound, it is often the case that only the first iteration successfully reduces the energy, and then the energy stays constant.

Still, the mean approximations are useful because they are *independent of permutations* σ. So, when permutations σ are not reliable (*e.g.*, when $S^t = \Omega$ at the initial iteration, S^t gives no clues for estimating σ), the use of the mean approximations is reasonable. Based on these observations, we in the next section propose a new bound that generalizes both of the semidifferentials in SSP [Narasimhan and Bilmes, 2005; Rother *et al.*, 2006] and the bounds in [Ayed *et al.*, 2013].

4.4 Proposed Method

4.4.1 Bound for Type-I

Given the non-linear function $R(S)$ in the form of Equation (4.4), we present our proposed bound for $R(S)$ by extending the semidifferentials in SSP [Narasimhan and Bilmes, 2005]. First, we define a *grouped permutation* π by grouping ordered-elements in $\sigma = \{\sigma(1), \sigma(2), \cdots, \sigma(|\Omega|)\}$ into M ($M \leq |\Omega|$) groups: $\pi(1), \pi(2), \cdots, \pi(M) \subseteq \Omega$. Each group $\pi(i)$ contains some consecutive elements of σ: $\pi(i) = \{\sigma(j), \sigma(j+1), \cdots, \sigma(j+m)\}$, and groups are mutually disjoint: $\pi(i) \cap \pi(j) = \emptyset$ if $i \neq j$. More importantly, any group does not across $\sigma(|S^t|)$ and $\sigma(|S^t|+1)$, thus there exists $S_j^\pi = S^t$ for some j. Using this grouped permutation π we define a maximal chain C^π: $S_0^\pi = \emptyset$ and $S_i^\pi = \pi(1) \cup \pi(2) \cup ... \cup \pi(i)$. The grouped permutation and corresponding chain are illustrated in Figure 4.2 (b). Then, our bound for $R(S)$ is defined similarly to that of SSP in Equation (4.18) as

$$
H^\pi(S|S^t) = \sum_{z \in Z} H_z^\pi(S|S^t) = \sum_{z \in Z} [\langle h_z^\pi, S \rangle + F_z(\emptyset)] \qquad (4.25)
$$

where unary costs $h_z^\pi : \Omega \to \mathbb{R}$ are defined for each $i \in \pi(j)$ as

$$h_z^\pi(i) = g_z(i) \left[F_z(S_j^\pi) - F_z(S_{j-1}^\pi) \right] / \langle g_z, \pi(j) \rangle \,. \tag{4.26}$$

Essentially, $H_z^\pi(S|S^t)$ is a piecewise-mean-approximation of SSP's bound $H^\sigma(S|S^t)$, as visualized in Figure 4.3 (c) using the example permutation given in Figure 4.2 (b).

Proposition: *The function $H^\pi(S|S^t)$ satisfies the conditions of Equation (4.13) and is thus a tight upper bound for $R(S)$ in Equation (4.4).*

Proof. To prove this, we show that $H_z^\pi(S|S^t)$ is an extreme point of $\partial F_z(S^t)$. From the definitions of π and C^π, there exists S_j^π such that $S_j^\pi = S^t$. The condition of Equation (4.17) holds for $H_z^\pi(S|S^t)$, since

$$\begin{aligned} H_z^\pi(S_j^\pi|S^t) - H_z^\pi(S_{j-1}^\pi|S^t) &= \langle h_z^\pi, \pi(j) \rangle \\ &= F_z(S_j^\pi) - F_z(S_{j-1}^\pi). \end{aligned}$$

□

Note that our bound $H^\pi(S|S^t)$ becomes equivalent to $H^\sigma(S|S^t)$ the semidifferentials in SSP, if $\pi(i) = \{\sigma(i)\}$. Moreover, it also becomes equivalent to the full-mean-approximation bound $H_{mean}(S|S^t)$ in Equation (4.24), if $\pi(1) = S^t$ and $\pi(2) = \Omega \setminus S^t$.

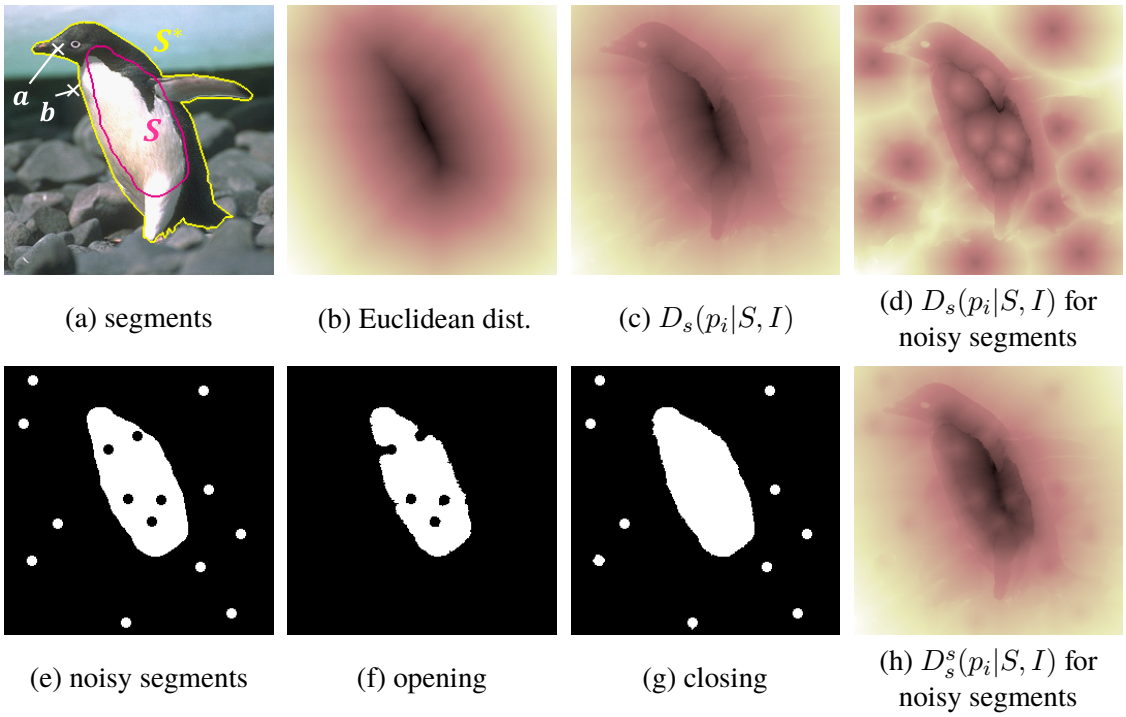
The spirit behind this grouping and piecewise-mean-approximation scheme is that, when the permutation of pixels $\sigma(i)$ and $\sigma(i+1)$ is expected to be unreliable, we put the two pixels into the same group in order to treat them equally and leave a decision (*i.e.*, whether which one is more likely to be foreground) to other interactions, *e.g.*, pairwise smoothness terms. How we make σ and π is described in the next section.

4.4.2 Geodesic Distance for Deciding Permutations

In [Rother *et al.*, 2006] permutations σ are made according to the signed distance from of the border of the current segmentation S^t. Here, we propose a more sophisticated method for deciding permutations by employing geodesic distance [Criminisi *et al.*, 2008].

Below we summarize the geodesic distance transform technique proposed in [Criminisi *et al.*, 2008] and describe how we use it in our method. A unsigned geodesic distance from given segments S is defined as

$$D(p_i|S, I) = \min_{\{p_j | j \in S\}} d_G(p_i, p_j), \tag{4.27}$$



(a) segments (b) Euclidean dist. (c) $D_s(p_i|S,I)$ (d) $D_s(p_i|S,I)$ for noisy segments

(e) noisy segments (f) opening (g) closing (h) $D^s_s(p_i|S,I)$ for noisy segments

Figure 4.4 Illustration of geodesic distance [Criminisi *et al.*, 2008]. Given (a) an image and segments (pink), the use of (c) geodesic distance $D_s(p_i|S,I)$ is more reasonable than (b) the Euclidean distance. But (d) $D_s(p_i|S,I)$ is sensitive to (e) noisy speckles in segments. With the presence of (f) opening and (g) closing effects, (h) $D^s_s(p_i|S,I)$ is robust to such speckles.

where $d_G(p_i,p_j)$ is a geodesic distance between two pixels $p_i, p_j \in \mathbb{R}^2$:

$$d_G(p_i,p_j) = \min_{s\in\mathcal{P}} \sum_{k=2}^{|s|} \sqrt{\|p_{s(k)} - p_{s(k-1)}\|_2^2 + \gamma^2 \|I_{p_{s(k)}} - I_{p_{s(k-1)}}\|_2^2}. \tag{4.28}$$

Here, $\mathcal{P}$ is the set of all paths joining p_i and p_j. Note that when $\gamma = 0$, the geodesic distance $d_G(p_i,p_j)$ becomes equivalent to the Euclidean distance $|p_i - p_j|$. Using the unsigned distance map, a signed geodesic distance from S is defined as

$$D_s(p_i|S,I) = D(p_i|S,I) - D(p_i|\bar{S},I). \tag{4.29}$$

On the one hand, using this signed geodesic distance for making permutations σ is reasonable. To illustrate this, see the two pixels a and b in Figure 4.4 (a). Using the Euclidean distance shown in Figure 4.4 (b), b has a shorter distance than a from S; hence, b is positioned before a in σ meaning that b is more likely to be within the true segments S^*. By contrast, when we use the geodesic distance shown in Figure 4.4 (c), it is likely that $D_s(a|S,I) < D_s(b|S,I)$, so a will come before b in σ reasonably.

On the other hand, as illustrated in Figure 4.4 (d), this distance is very sensitive to noise speckles in S, *i.e.*, small holes in foreground S and background $\bar{S}$ such shown in Figure 4.4 (e).

For this issue, the authors of [Criminisi *et al.*, 2008] further introduce *dilation* and *erosion* for segments S, which may be defined respectively as

$$S_d = \{i \in \Omega \,|\, D_s(p_i|S, I) \leq +\theta_d\}, \tag{4.30}$$
$$S_e = \{i \in \Omega \,|\, D_s(p_i|S, I) \leq -\theta_e\}, \tag{4.31}$$

Then, *opening* and *closing* for S are defined respectively as

$$S_o = \{i \in \Omega \,|\, D(p_i|S_e, I) \leq +\theta_e\}, \tag{4.32}$$
$$S_c = \{i \in \Omega \,|\, D(p_i|\bar{S}_d, I) > +\theta_d\}. \tag{4.33}$$

The effects of opening and closing are illustrated in Figures 4.4 (f) and (g). Based on these operations, a signed distance with opening and closing effects is given by

$$D_s^s(p_i|S, I) = [D(p_i|S_e, I) - \theta_e] - \left[D(p_i|\bar{S}_d, I) - \theta_d\right]. \tag{4.34}$$

The effect of this distance is visualized in Figure 4.4 (h). The parameters θ_d and θ_e reflect the maximum sizes of speckles in foreground S and background $\bar{S}$, respectively. We use $\theta_d = \theta_e = 5$ and $\gamma = 10/255$.

Equipped with this signed distance the construction of the bound function $H^\pi(S|S^t)$ is as follows. First, we compute this signed distance $D_s^s(p_i|S^t, I)$ for the current segments S^t. Second, we make a permutation σ according to this distance such that $D_s^s(p_{\sigma(i)}|S^t, I) \leq D_s^s(p_{\sigma(i+1)}|S^t, I)$. Finally, we make a grouped permutation π from σ. We process $\sigma(i)$ from $\sigma(2)$ to $\sigma(|\Omega|)$, and put $\sigma(i)$ into the same group with $\sigma(i-1)$ if $D_s^s(p_{\sigma(i)}|S^t, I) - D_s^s(p_{\sigma(i-1)}|S^t, I) \leq \tau$ and not if $\sigma(i-1) \in S^t$ and $\sigma(i) \in \bar{S}^t$. See also the supplementary for the implementation details. Basically, the size of a threshold τ reflects how much the permutation σ and so the distance $D_s^s(p_i|S^t, I)$ are reliable. We empirically use a grouping threshold given by $\tau = a/t^3 \geq 0$ $(t = 1, 2, 3, ...)$ because as iterations proceed the segments S^t are expected to be more accurate and so permutations σ by the distance from S^t becomes accordingly more reasonable. Also, when $S^t = \Omega$ or $S^t = \emptyset$ at the first iteration, so the distance from S^t cannot be defined, we set $\pi(1) = \Omega$. This makes the full linear approximations of $F_z(S)$ drawn from $F_z(\emptyset)$ to $F_z(\Omega)$, which correspond to the bound $B_{shrink}(S|S^t)$ of [Ayed *et al.*, 2013] and give reasonable approximations for the initial start-up.

4.4.3 Bound for Type-II

Following the derivation of [Ayed *et al.*, 2013], we can derive a bound for the Type-II form of $R(S)$, with the tolerance of losing the bidirectionality. Since f_z is a monotonically decreasing function, a bound $\hat{R}(S)$ for $R(S)$ may be derived using an auxiliary variable $S \subseteq S^t$ as

$$R(S) = \sum_{z \in Z} f_z\left(\frac{\langle g_z, S\rangle}{\langle w_z, S\rangle}\right) \leq \sum_{z \in Z} f_z\left(\frac{\langle g_z, S\rangle}{\langle w_z, S^t\rangle}\right) = \hat{R}(S), \tag{4.35}$$

which is now the Type-I form of Equation (4.4). So we can further derive a bound $H^{\pi}(S \subseteq S^t|S^t)$ for $\hat{R}(S)$. The restriction $S \subseteq S^t$ can be achieved by giving hard constraints to $i \in \Omega \setminus S^t$. To avoid over-shrinking, we append the α-term of Equation (4.21) to $H^{\pi}(S \subseteq S^t|S^t)$, and further add the smoothness term $Q(S)$ to make a overall bound $\hat{E}(S|S^t, \alpha)$ for $E(S)$.

4.5 Experimental Evaluations

We evaluate the performances of our method, SSP [Narasimhan and Bilmes, 2005; Rother *et al.*, 2006], and AC [Ayed *et al.*, 2013], namely, by comparing the following four methods:

SC-GEO is the our proposed method described in Section 4.4.

SC-DIST uses the standard Euclidean distance (*i.e.*, $\theta_e = \theta_d = \gamma = 0$) for making permutations σ, but the other settings are the same with SC-GEO.

SSP-DIST is SSP [Narasimhan and Bilmes, 2005] implemented following the descriptions in [Rother *et al.*, 2006]. Basically, SSP-DIST is the same with SC-DIST but uses no mean approximations. When $S^0 = \Omega$ at the first iteration, a permutation σ is made randomly based on 10×10-pixels of patches as described in [Rother *et al.*, 2006].

AC is the method proposed in [Ayed *et al.*, 2013], which uses the bound of Equation (4.21).

4.5.1 The GrabCut Benchmark Evaluations using L_1 and L_2-Distances

Similarly to [Ayed *et al.*, 2013; Rother *et al.*, 2006], we evaluate the performances of the four methods using the GrabCut dataset [Rother *et al.*, 2004]; given the target histogram of the ground truth segmentation we compare the four methods using the L_2 and L_1-distance measures of histograms. We use RGB-histograms of 192^3 and 64^3 bins. We use the following form of 16-neighborhood smoothness term:

$$Q(S) = \lambda \sum_{(i,j)\in N} \max(\exp(-\beta|I_{p_i} - I_{p_j}|^2, \epsilon)/|p_i - p_j|\, \delta(\chi_S(i) - \chi_S(j)), \tag{4.36}$$

where $\beta = (2E[|I_{p_i} - I_{p_j}|^2])^{-1}$ is computed via the expectation over the image. For $\{\lambda, \epsilon\}$, we use $\{1.0, 0.5\}$ and $\{0.5, 0.5\}$ for the L_2 and L_1-distances, respectively. S^t is trivially initialized as $S^0 \leftarrow \Omega$ but for SC-GEO and AC we also use the results of BJGC [Boykov and Jolly, 2001] as $S^0 \leftarrow S^{BJ}$ to show the useful properties of our method. For both SC-GEO and SC-DIST, we use a grouping threshold $\tau = 300/t^3$ for L_2 and $\tau = 10/t^3$ for L_1. When using BJGC initialization, so S^t is relatively accurate from the beginning, we use $\tau = 10/t^3$ for both L_2 and L_1. For AC, we use $\alpha = 0.5$.

Tables 4.1 and 4.2 summarize the performances using the L_2 and L_1-distances, showing average misclassified pixel rates and energy values of $E(S)$ and $R(S)$, and individual-image comparisons with SC-GEO. Among the four methods, our proposed method SC-GEO outperforms the others for all settings. Notably, SC-GEO shows about an order of magnitude greater accuracies than AC [Ayed *et al.*, 2013] beating for almost all individual images. Comparing the results of SC-DIST and SSP-DIST with L_2 and 64^3 bins, SC-DIST finds more accurate segmentations in spite of the higher energies. This is because in SSP-DIST the appearance consistencies are forced regardless of how permutations σ and corresponding bounds are accurate, resulting in highly non-smooth, visibly bad local minimas. Figures 4.5 and 4.6 show the plots of the energy value transitions $E(S^t)$ using the L_2 and L_1-distances, where the energies $E(S^t)$ are averaged over the 50 images. As shown, SC-GEO achieves greater convergence than SSP-DIST and AC in all settings. Also, the use of plausible initialization is effective for our method promoting the convergence, but not much effective for AC. This is because AC can only find segments by reducing the initial segments.

4.6 Application to Multiple Distribution Matching

4.6.1 Formulation

Multiple distribution matching problems were first addressed in [Taniai *et al.*, 2012] particularly for $K = 2$ with the Bhattacharyya coefficient. A notable point of their work is that they have derived appropriate weights for the two distribution matching terms, by which the method becomes significantly robust when using approximate input distributions. In this paper, we extend their result and derive more general formulations assuming $K \geq 2$ with the L_1-distance as well as the Bhattacharyya coefficient.

Multiple Bhatthacharyya Models

Let $\Pr(z|S)$ be the pixel feature distribution in segments S defined as

$$\Pr(z|S) = \frac{\langle k_z, S \rangle}{\langle 1, S \rangle}, \tag{4.37}$$

where $k_z : \Omega \to \mathbb{R}$ is some kernel function. We consider the following form of appearance term

$$R(D_K) = -\sum_{i=1}^{K} \sum_{z \in Z} \lambda_i \sqrt{\Pr(z|S_i)\mathcal{H}_i(z)} \tag{4.38}$$

$$= \sum_{i=1}^{K} \lambda_i R_i(S_i|\mathcal{H}_i) \tag{4.39}$$

Table 4.1 Evaluations on the GrabCut benchmark [Rother *et al.*, 2004] using L_2-distance. We show average error rates, $E(S)$, $R(S)$ over 50 images. The last column shows the number of images for which SC-GEO outperforms. We compare our **SC-GEO** and SC-DIST with SSP-DIST [Rother *et al.*, 2006] and AC [Ayed *et al.*, 2013]. We use 192^3 and 64^3 bins, and two types of initialization (all-region and BJGC [Boykov and Jolly, 2001]).

Method ($\mathbf{L_2}$**-distance**)	Init.	Error (%)		$E(S)$		$R(S)$		SC-GEO vs	
		192^3	64^3	192^3	64^3	192^3	64^3	192^3	64^3
Ground Truth	-	0	0	3569	3569	0	0	-	-
SC-GEO	all	**0.106**	**0.380**	**3538**	**4090**	**196**	**341**	-	-
SC-DIST	all	0.141	1.017	4163	20347	610	12464	34	45
SSP-DIST	all	0.402	2.566	4850	14060	349	874	33	50
AC	all	1.256	3.278	20403	178151	15165	164191	50	50
SC-GEO	BJGC	**0.100**	**0.353**	**3506**	**3972**	**178**	**317**	-	-
AC	BJGC	0.615	0.906	(5166)	(20731)	2120	17598	50	50
ref. BJGC	-	0.802	1.000	(5707)	(21683)	(2663)	(18632)	-	-

Table 4.2 Evaluations on the GrabCut benchmark [Rother *et al.*, 2004] using L_1-distance. We show average error rates, $E(S)$, $R(S)$ over 50 images. The last column shows the number of images for which SC-GEO outperforms. We compare our **SC-GEO** and SC-DIST with SSP-DIST [Rother *et al.*, 2006] and AC [Ayed *et al.*, 2013]. We use 192^3 and 64^3 bins, and two types of initialization (all-region and BJGC [Boykov and Jolly, 2001]).

Method ($\mathbf{L_1}$**-distance**)	Init.	Error (%)		$E(S)$		$R(S)$		SC-GEO vs	
		192^3	64^3	192^3	64^3	192^3	64^3	192^3	64^3
Ground Truth	-	0	0	1785	1785	0	0	-	-
SC-GEO	all	**0.033**	**0.283**	**1790**	**1923**	**37**	**151**	-	-
SC-DIST	all	0.039	0.307	1802	2009	58	257	42	37
SSP-DIST	all	0.044	0.676	1807	2342	43	267	42	44
AC	all	0.399	1.292	2620	4336	877	2475	50	49
SC-GEO	BJGC	**0.033**	**0.245**	**1790**	**1901**	**41**	**151**	-	-
AC	BJGC	0.517	0.874	2603	3179	1006	1626	50	50
ref. BJGC	-	0.802	1.000	(3134)	(3434)	(1612)	(1908)	-	-

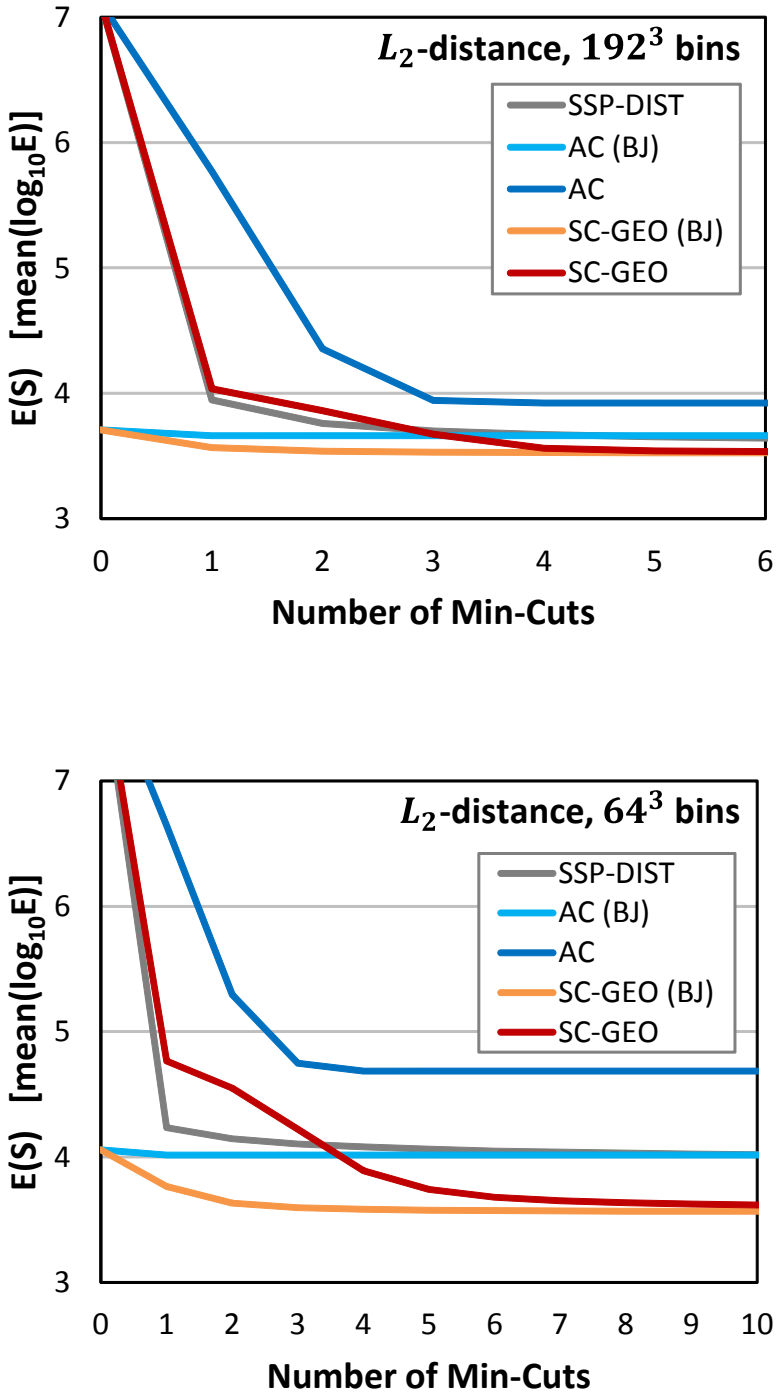


Figure 4.5 Energy convergence comparisons using the L_2-distance with 192^3 and 64^3 bins. The energy function values $E(S)$ w.r.t. the number of min-cuts (*i.e.* iterations) are shown. Our method (**SC-GEO**) significantly outperforms both AC [Ayed *et al.*, 2013] and SSP-DIST [Narasimhan and Bilmes, 2005; Rother *et al.*, 2006]. The use of plausible initialization (BJ) is effective and promotes the convergence of our method but not for AC [Ayed *et al.*, 2013].

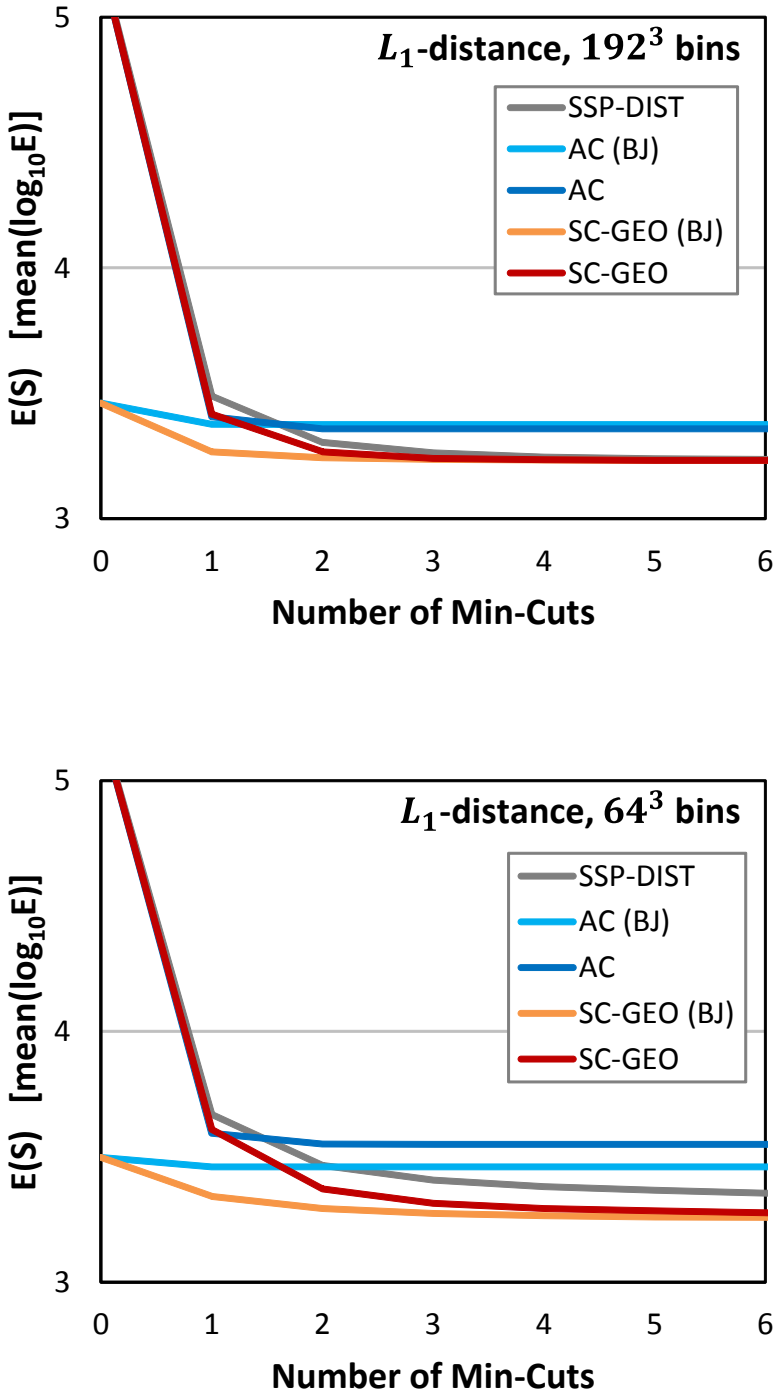


Figure 4.6 Energy convergence comparisons using the L_1-distance with 192^3 and 64^3 bins. The energy function values $E(S)$ w.r.t. the number of min-cuts (*i.e.* iterations) are shown. Our method (**SC-GEO**) significantly outperforms both AC [Ayed *et al.*, 2013] and SSP-DIST [Narasimhan and Bilmes, 2005; Rother *et al.*, 2006]. The use of plausible initialization (BJ) is effective and promotes the convergence of our method but not for AC [Ayed *et al.*, 2013].

where $\mathcal{H}_i(z) := \Pr(z|\mathcal{O}_i)$ is a priori known probability distribution for the i-th object $\mathcal{O}_i$. Following the derivation of [Taniai *et al.*, 2012] we derive the weights λ_i via *matching of entire image distributions*.

Let $\mathcal{H}_\Omega(z) := \Pr(z|\Omega)$ be the distribution in the entire image region. We approximately express $\mathcal{H}_\Omega(z)$ by the sum of input distributions as

$$\tilde{\mathcal{H}}_\Omega(z|\boldsymbol{\eta}) = \sum_{i=1}^{K} \eta_i \mathcal{H}_i(z), \tag{4.40}$$

where $\eta_i \geq 0$ and $\sum_i \eta_i = 1$. Then, we define the Bhattacharyya coefficient between $\mathcal{H}_\Omega(z)$ and $\tilde{\mathcal{H}}_\Omega(z|\boldsymbol{\eta})$:

$$R_\Omega(\boldsymbol{\eta}) = -\sum_{z \in Z} \sqrt{\mathcal{H}_\Omega(z)\tilde{\mathcal{H}}_\Omega(z|\boldsymbol{\eta})} \tag{4.41}$$

This function $R_\Omega(\boldsymbol{\eta})$ can be shown to be convex, therefore its minimizer

$$\boldsymbol{\eta}^* = \arg\min R_\Omega(\boldsymbol{\eta}) \tag{4.42}$$

can be obtained by *e.g.* gradient descents. Here, η_i^* represents an estimate for the ratio of the object-i's region size to the image size $|S_i^*|/|\Omega|$. When input distributions are given as the true distributions, $\eta_i^* = |S_i^*|/|\Omega|$ as well as $\mathcal{H}_\Omega(z) = \tilde{\mathcal{H}}_\Omega(z|\boldsymbol{\eta}^*)$ holds.

On the other hand, an upper bound for $R_\Omega(\boldsymbol{\eta}^*)$ can be derived as

$$R_\Omega(\boldsymbol{\eta}^*) \leq \sum_{i=1}^{K} \sqrt{\eta_i^* \frac{\langle 1, S_i \rangle}{|\Omega|}} R_i(S_i|\mathcal{H}_i) \tag{4.43}$$

$$= -\sum_{i=1}^{K} \sqrt{\eta_i^* \frac{1}{|\Omega|}} \sum_{z \in Z} \sqrt{\langle k_z, S_i \rangle \, \mathcal{H}_i(z)} \tag{4.44}$$

which holds equal if $\eta_i^* = |S_i^*|/|\Omega|$ and $\Pr(z|S_i) = \mathcal{H}_i(z)$. Equation (4.43) means that *the energy* $R_\Omega(\boldsymbol{\eta}^*)$ *that measures the similarity between true and approximate entire-image-distributions is closely connected with the energy that measures similarities between individual model distributions* $\Pr(z|S_i)$ *and* $\mathcal{H}_i(z)$. Therefore, we use the right-hand side of Equation (4.43) as the appearance term for multi-model segmentation. But because the term is bounded in $[-1, 0]$, we use a $|\Omega|$-factored term to account for the balance with $Q(D_K)$:

$$R(D_K) = -\sum_{i=1}^{K} \sqrt{\eta_i^* |\Omega|} \sum_{z \in Z} \sqrt{\langle k_z, S_i \rangle \, \mathcal{H}_i(z)} \tag{4.45}$$

Seeing Equation (4.43) the physical meaning for the weights of $R_i(S_i|\mathcal{H}_i)$ obtained here is very intuitive and reasonable; namely, each matching term $R_i(S_i|\mathcal{H}_i)$ should be weighted by the geometric mean of area size ratios computed by two ways, *i.e.*, from prior distributions

and from resulting segments. Also, from Equation (4.44), the normalization terms $\langle 1, S_i \rangle$ in $\Pr(z|S_i)$ are canceled and each matching term becomes the Type-I form of Equation (4.4). This formulation allows us to use bidirectional optimization and thus allows the use of expansion and swap algorithms.

Multiple L_1-Distance Models

Similarly to the case of the Bhattacharyya measures, we can derive multiple distribution matching models for L_1-distance:

$$R(D_K) = \sum_{i=1}^{K} R_i(S_i|\mathcal{H}_i) \tag{4.46}$$

$$= \sum_{i=1}^{K} \sum_{z \in Z} \left| \langle k_z, S_i \rangle - \eta_i^* |\Omega| \mathcal{H}_i(z) \right|, \tag{4.47}$$

where η_i^* are defined as the minimizer of the L_1-distance between $\mathcal{H}_\Omega(z)$ and $\tilde{\mathcal{H}}_\Omega(z|\boldsymbol{\eta})$, just like the case of the Bhattacharyya models.

4.6.2 Expansion and Swap Algorithms

For $K = 2$, the appearance term $R(D_2) = R_1(S_1|\mathcal{H}_1) + R_2(S_2|\mathcal{H}_2)$ can be expressed by $R(S) = R_1(S|\mathcal{H}_1) + R_2(\Omega \setminus S|\mathcal{H}_2)$, which is essentially binary segmentation problems and thus our SC-GEO algorithm can be directly applied.

For $K > 2$, we use well-known α-expansion and $\alpha\beta$-swap algorithms [Boykov *et al.*, 2001]. The application of expansion and swap algorithms is straightforward. In each $\alpha/\alpha\beta$-move we iteratively minimize the bounds of the move energies, which is achieved as a direct extension of the SC-GEO algorithm. In α-expansion, since we expand the region $S_{i=\alpha}$ and shrink the others, we set terminal-edge costs as $t(i) \leftarrow h^\pi(i|l = \alpha)$ and source-edge costs as $s(i) \leftarrow h^\pi(i \in S_k^t|l = k)$, where $h^\pi(i|l)$ is unary costs of the object-l's bound $H^\pi(S_l|S_l^t, l)$. In $\alpha\beta$-swap, since two regions $S_{i=\alpha}$ and $S_{i=\beta}$ can exchange their pixels while the others are fixed, we set terminal-edge costs as $t(i) \leftarrow h^\pi(i|l = \alpha)$ and source-edge costs as $s(i) \leftarrow h^\pi(i|l = \beta)$ for $i \in S_\alpha \cup S_\beta$. Also, when computing $H^\pi(S_l|S_l^t, l)$, $l = \alpha, \beta$, we set a sufficiently large value d_{max} to $D_s^s(p_i|S_l^t, I)$ where $i \in \Omega \setminus (S_\alpha^t \cup S_\beta^t)$, because such pixels i are never labeled as α nor β.

Notice that the bidirectionality of optimization is essential here to use expansion and swap algorithms because they contain both two ($shrink$ and $expand$) directions of optimization in each move.

4.6.3 Results

We first compare our method with [Taniai *et al.*, 2012] using the Battacharyya measures with $K = 2$. We can directly use the proposed SIC-GEO method described in Section 4.4. We use the

smoothness term and parameters specified in [Taniai *et al.*, 2012] and true distributions learned from ground truth. Since the algorithm of [Taniai *et al.*, 2012] internally uses BJGC's results in optimization, we also use them as the initial segments. Table 4.3 shows performance comparisons on the GrabCut benchmark. Our method achieves greater accuracy in smaller numbers of min-cut operations.

Since the bound used in [Taniai *et al.*, 2012] is essentially the same with that of AC [Ayed *et al.*, 2013], the method [Taniai *et al.*, 2012] uses two auxiliary labels $S^t_{\mathcal{F}}$ and $S^t_{\mathcal{B}}$ for foreground and background terms, and alternately optimize the two terms, which makes their optimization procedure complicated. The bidirectinal optimization of our method leads to a much simplified and straightforward procedure.

To evaluate the performance for general $K > 2$ cases, we use six images from the BSDS500 dataset [Arbelaez *et al.*, 2011]. We use the smoothness term of Equation (4.36) and set $\lambda = \epsilon = 0.5$, $\tau = 300/t^3$ for the L_1-distance model, but $\lambda = 0.1$ for the Bhattacharyya model. The other parameters are set to the default. We also compare with the BJ model [Boykov and Jolly, 2001], for which we use $\lambda = 4$ and $\epsilon = 0.4$. We use 128^3 bins, and distributions as well as area-size rates η^*_i are learned from ground truth.

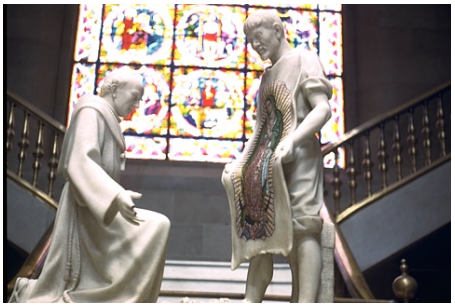
Table 4.4 shows the error rates of the six images using the three models with expansion and swap algorithms. All of our methods significantly outperform the BJ method. In Figures 4.7–4.12 we show the results for the six images, respectively. Our methods work quite well even for difficult camouflage scenes and thin structures. See also the supplementary for the complete results.

Table 4.3 Evaluations on the GrabCut benchmark [Rother *et al.*, 2004] using a dual Bhattacharyya model [Taniai *et al.*, 2012]. We show error rates and the number of min-cuts averaged over 50 images.

Method	Error (%)		#min-cuts	
	192^3	64^3	192^3	64^3
SC-GEO	**0.023**	**0.212**	3.7+1	6.3+1
DDM [Taniai *et al.*, 2012]	0.213	0.545	10.6+1	10.8+1
ref. BJGC [Boykov and Jolly, 2001]	0.802	1.000	1	1

Table 4.4 Evaluations of multiple distribution matching. Error rates (%) for six images in the BSDS500 dataset are shown. We use 128^3 bins, and true histograms learned from ground truth. We compare our four methods with the BJ model [Boykov and Jolly, 2001].

Model	Algorithm	**24077**	**41025**	**87046**	**106024**	**299086**	**335088**	Average
L_1-distance	expansion	1.88	1.20	0.40	0.13	0.26	0.42	0.72
L_1-distance	swap	1.77	1.08	**0.30**	**0.11**	0.26	**0.31**	0.64
Bhattacharyya	expansion	**1.70**	1.00	0.50	0.13	**0.24**	0.39	0.66
Bhattacharyya	swap	1.89	**0.58**	0.33	0.12	**0.24**	0.46	**0.60**
BJ model	expansion	6.85	3.69	2.72	2.13	1.39	2.34	3.19
BJ model	swap	6.85	3.62	2.72	2.16	1.39	2.37	3.19



24077

ground truth ($K = 5$)

L_1-distance (expansion)
$\epsilon = 1.88\%$

L_1-distance (swap)
$\epsilon = \underline{1.77}\%$

Bhattacharyya (expansion)
$\epsilon = \mathbf{1.70}\%$

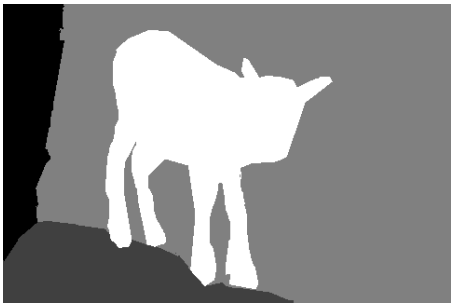
Bhattacharyya (swap)
$\epsilon = 1.89\%$

Boykov and Jolly [2001] (expansion)
$\epsilon = 6.85\%$

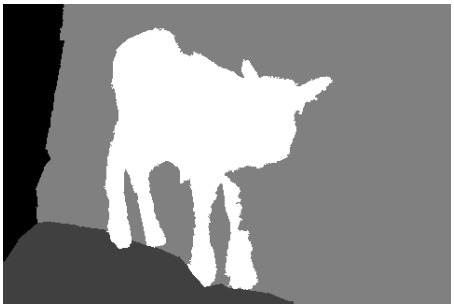
Boykov and Jolly [2001] (swap)
$\epsilon = 6.85\%$

Figure 4.7 Results of multiple distribution matching for "24077". We show the results of four proposed methods and the results using the BJ model [Boykov and Jolly, 2001].

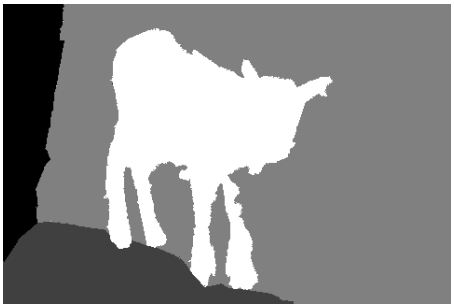
41025



ground truth ($K = 3$)



L_1-distance (expansion)
$\epsilon = 1.20\%$



L_1-distance (swap)
$\epsilon = 1.08\%$

Bhattacharyya (expansion)
$\epsilon = \underline{1.00}\%$

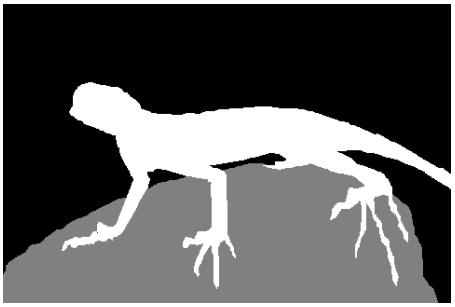
Bhattacharyya (swap)
$\epsilon = \mathbf{0.58}\%$

Boykov and Jolly [2001] (expansion)
$\epsilon = 3.69\%$

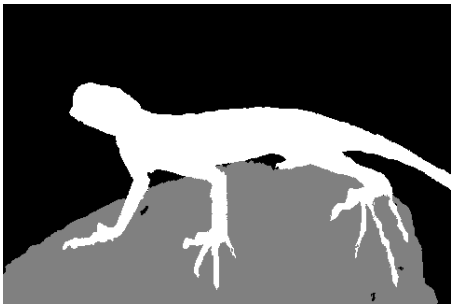
Boykov and Jolly [2001] (swap)
$\epsilon = 3.62\%$

Figure 4.8 Results of multiple distribution matching for "41025". We show the results of four proposed methods and the results using the BJ model [Boykov and Jolly, 2001].

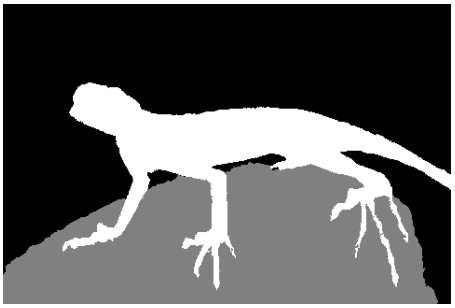
87046



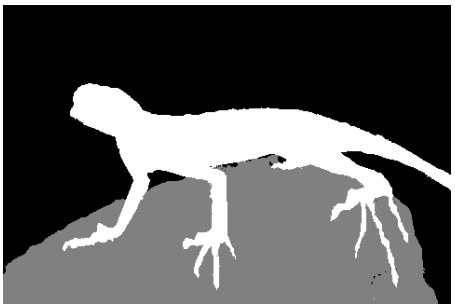
ground truth ($K = 3$)



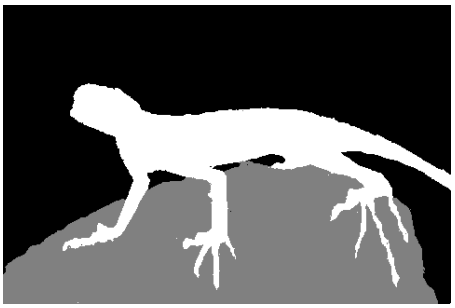
L_1-distance (expansion)
$\epsilon = 0.40\%$



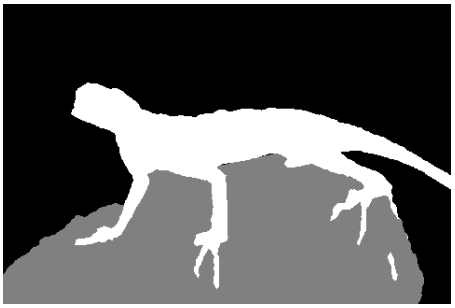
L_1-distance (swap)
$\epsilon = \mathbf{0.40}\%$



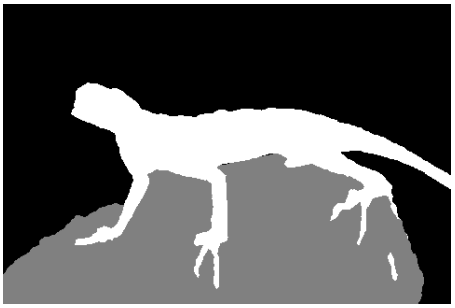
Bhattacharyya (expansion)
$\epsilon = 0.50\%$



Bhattacharyya (swap)
$\epsilon = \underline{0.33}\%$

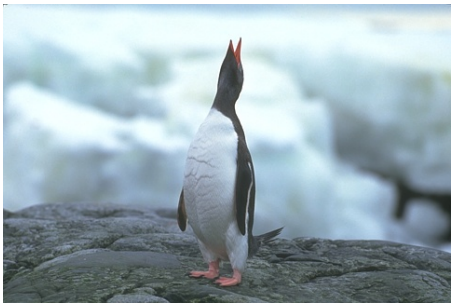


Boykov and Jolly [2001] (expansion)
$\epsilon = 2.72\%$

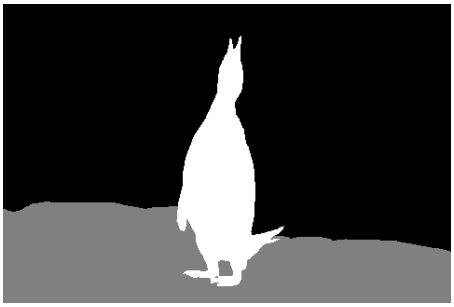


Boykov and Jolly [2001] (swap)
$\epsilon = 2.72\%$

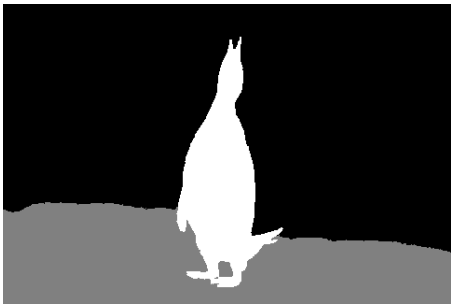
Figure 4.9 Results of multiple distribution matching for "87046". We show the results of four proposed methods and the results using the BJ model [Boykov and Jolly, 2001].



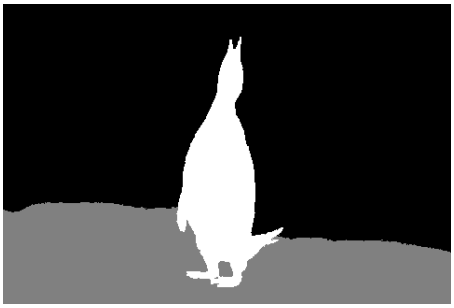
106024



ground truth ($K = 3$)

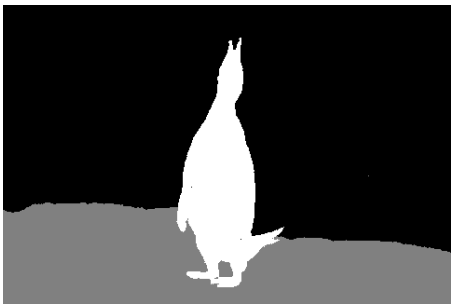


L_1-distance (expansion)
$\epsilon = 0.13\%$



L_1-distance (swap)
$\epsilon = \mathbf{0.11}\%$

Bhattacharyya (expansion)
$\epsilon = 0.13\%$



Bhattacharyya (swap)
$\epsilon = \underline{0.12}\%$

Boykov and Jolly [2001] (expansion)
$\epsilon = 2.13\%$

Boykov and Jolly [2001] (swap)
$\epsilon = 2.16\%$

Figure 4.10 Results of multiple distribution matching for "106024". We show the results of four proposed methods and the results using the BJ model [Boykov and Jolly, 2001].

299086

ground truth ($K = 5$)

L_1-distance (expansion)
$\epsilon = 0.26\%$

L_1-distance (swap)
$\epsilon = 0.26\%$

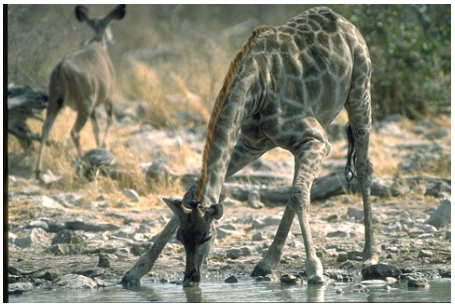
Bhattacharyya (expansion)
$\epsilon = \mathbf{0.24}\%$

Bhattacharyya (swap)
$\epsilon = \mathbf{0.24}\%$

Boykov and Jolly [2001] (expansion)
$\epsilon = 1.39\%$

Boykov and Jolly [2001] (swap)
$\epsilon = 1.39\%$

Figure 4.11 Results of multiple distribution matching for "299086". We show the results of four proposed methods and the results using the BJ model [Boykov and Jolly, 2001].

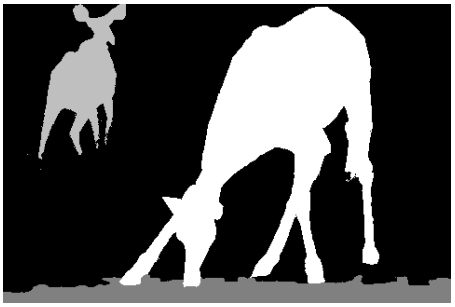


335088

ground truth ($K = 4$)

L_1-distance (expansion)
$\epsilon = \underline{0.42}\%$

L_1-distance (swap)
$\epsilon = \mathbf{0.31}\%$



Bhattacharyya (expansion)
$\epsilon = 0.39\%$

Bhattacharyya (swap)
$\epsilon = 0.46\%$

Boykov and Jolly [2001] (expansion)
$\epsilon = 2.34\%$

Boykov and Jolly [2001] (swap)
$\epsilon = 2.37\%$

Figure 4.12 Results of multiple distribution matching for "335088". We show the results of four proposed methods and the results using the BJ model [Boykov and Jolly, 2001].

4.7 Application to One-Cut Segmentation

One of the major drawbacks of distribution matching approaches is expensive computational costs; they are usually thought to require many min-cut operations to obtain approximate solutions. Contrary to such thoughts, we show here that our method can produce approximate solutions in only a single min-cut. To achieve this, we assume that sparse user-scribbles annotating foreground and background are additionally given. See Figure 4.13 for examples of such user-scribbles. The use of such information is very reasonable and common in interactive segmentation [Boykov and Jolly, 2001; Price *et al.*, 2010; Bai and Sapiro, 2007].

4.7.1 Geodesic Distance for User-Scribbles

The method presented here is essentially the same with the one described in Section 4.4. The only difference is the definition of geodesic distance, for which we use another type of geodesic distance used in [Bai and Sapiro, 2007; Price *et al.*, 2010] that is suitable for sparse user-scribbles.

Let $\Omega_{\mathcal{F}}, \Omega_{\mathcal{B}} \subset \Omega$ be user-scribbled pixels annotating foreground and background (*i.e.*, red and blue pixels in Figure 4.13), respectively. Following [Bai and Sapiro, 2007; Price *et al.*, 2010] we define a new geodesic distance from scribbles $\Omega_{\mathcal{F}}$ or $\Omega_{\mathcal{B}}$ as

$$D_l(p_i|\Omega_l, I) = \min_{\{p_j|j\in\Omega_l\}} d_l(p_i, p_j), \ \ (l = \mathcal{F}, \mathcal{B}), \tag{4.48}$$

where a geodesic distance $d_l(p_i, p_j)$ between two pixels p_i, p_j is defined as

$$d_l(p_i, p_j) = \min_{s\in\mathcal{P}} \sum_{k=2}^{|s|} |P_l(I_{p_{s(k)}}) - P_l(I_{p_{s(k-1)}})|, \ \ (l = \mathcal{F}, \mathcal{B}). \tag{4.49}$$

Here, $\mathcal{P}$ is the set of all paths joining p_i and p_j, and $P_l(z)$ is defined as

$$P_l(z) = \frac{\Pr(z|l)}{\Pr(z|\mathcal{F}) + \Pr(z|\mathcal{B})}, \ \ (l = \mathcal{F}, \mathcal{B}). \tag{4.50}$$

Intuitively, the distance $D_l(p_i|\Omega_l, I)$ represents some (dis-)likelihoods spatially propagated from scribbles Ω_l. By measuring the distance using the gradients of the relative probability distribution $P_l(I_p)$ rather than intensities I_p themselves, it becomes robust to textures and noises. Like [Price *et al.*, 2010], we further define a relative foreground/background geodesic distance by normalizing $D_{\mathcal{F}}(p_i|\Omega_l, I)$ and $D_{\mathcal{B}}(p_i|\Omega_l, I)$ as

$$D(p_i|\Omega_{\mathcal{F}}, \Omega_{\mathcal{B}}, I) = \frac{D_{\mathcal{F}}(p_i|\Omega_l, I)}{D_{\mathcal{F}}(p_i|\Omega_l, I) + D_{\mathcal{B}}(p_i|\Omega_l, I)}. \tag{4.51}$$

Essentially, its negative form $\bar{D}(p_i|\Omega_{\mathcal{F}}, \Omega_{\mathcal{F}}, I) = 1 - D(p_i|\Omega_{\mathcal{F}}, \Omega_{\mathcal{F}}, I)$ represents likelihoods of pixels p_i being foreground as visualized in Figure 4.13.

In our one-cut segmentation, in stead of using $D_s^s(p_i|S, I)$ in Equation (4.34), we use $D(p_i|\Omega_{\mathcal{F}}, \Omega_{\mathcal{B}}, I)$ for making a permutation σ. Then, we make a grouped permutation π from σ by using a threshold $\tau = 10^{-8}$. Our one-cut segmentation is then achieved by performing a single min-cut for a bound $H^{\pi}(S|S^0 = \Omega) + Q(S)$.

4.7.2 Results

We evaluate our one-cut segmentation method using the GrabCut dataset [Rother *et al.*, 2004] and user-scribbles provided in [Gulshan *et al.*, 2010]. To assess the pure performance of the proposed optimization scheme, we use no hard constraints for scribble pixels $\Omega_{\mathcal{F}}$, $\Omega_{\mathcal{B}}$, and we use histograms learned from ground truth. The smoothness term and its parameters are the same with the default settings specified in Section 4.5.1.

Table 4.5 summarizes the performance of our method using L_1 and L_2-distance measures, showing approximate solutions are indeed obtained in one min-cut. In Figure 4.13, we show some example results of our one-cut method (SC-OneCut) with the corresponding results of geodesic segmentation [Bai and Sapiro, 2007], which are obtained by thresholding the geodesic distance map $D(p_i|\Omega_{\mathcal{F}}, \Omega_{\mathcal{F}}, I)$ using a threshold 0.5. Seeing the results of the "scissor" and "swimmer" examples, geodesic segmentation [Bai and Sapiro, 2007] cannot correctly label isolated regions because likelihoods cannot be propagated from the scribbles to the isolated regions. By contrast, our method uses the geodesic distance likelihoods only for making permutations σ, π and for constructing an appropriate bound function, thus it is robust to such issues. We further compare the performances of our method described in Section 4.4 (SC-GEO) and this one-cut method. In Figures 4.14–4.17, we show the segmentation results and the plots of the energy function values $E(S)$ of SC-OneCut and SC-GEO for some difficult "camouflage" examples. As shown, SC-GEO finds very accurate solutions, but the convergence is sometimes slow for such camouflage images. In spite of such difficulties, SC-OneCut can find good approximate solutions in one min-cut and even outperforms SC-GEO in the "grave" example.

Table 4.5 Evaluations of the proposed one-cut segmentation on the GrabCut benchmark [Rother *et al.*, 2004]. We show average error rates, $E(S)$, and $R(S)$ over 50 images. Our method yields good approximate solutions in only a single min-cut.

Model	Error (%)		$E(S)$		$R(S)$	
	192^3	64^3	192^3	64^3	192^3	64^3
L_1-distance	0.127	0.585	1959	2460	201	673
L_2-distance	0.211	0.711	3909	6047	412	1383

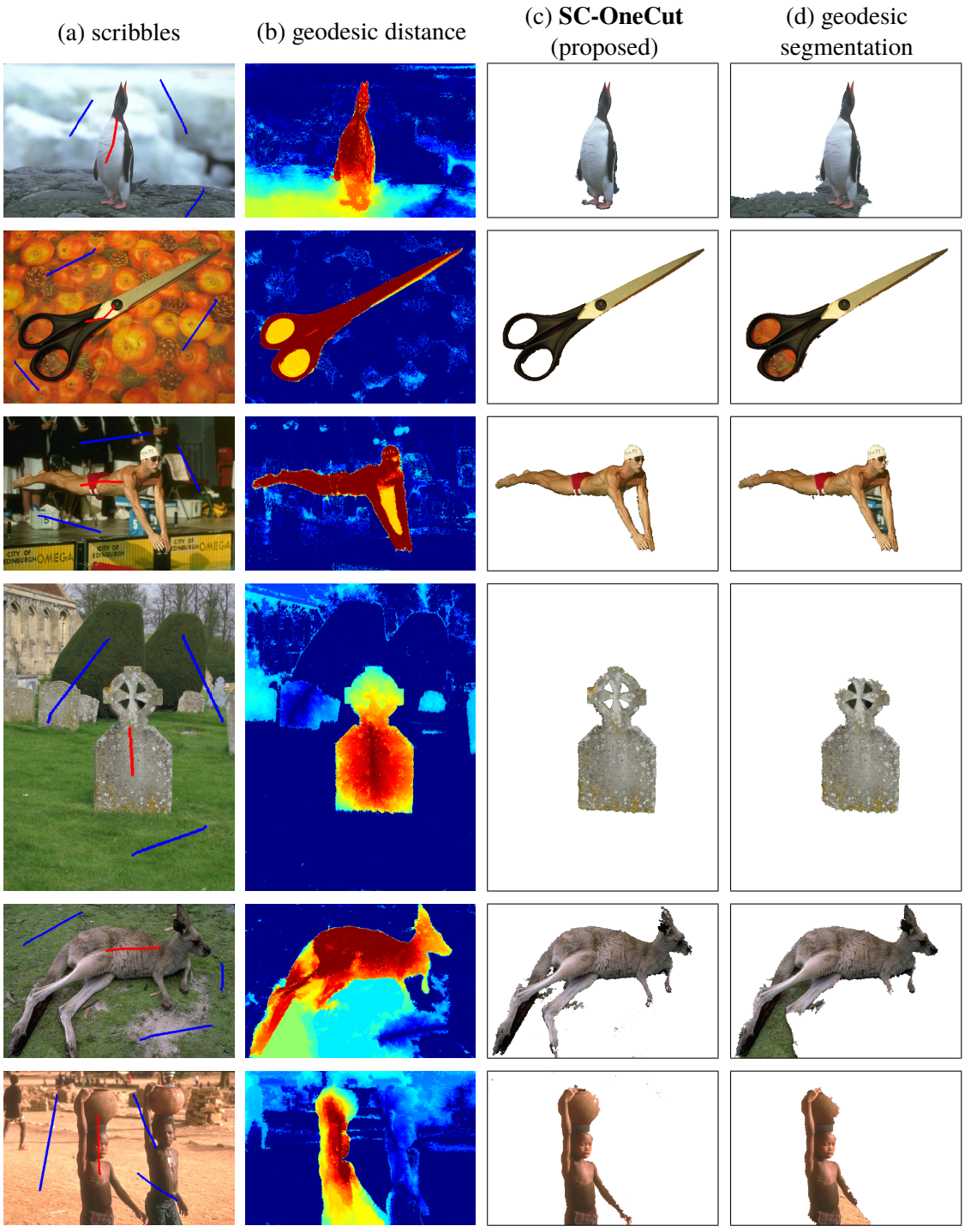


Figure 4.13 Example results of our one-cut segmentation. From left to right, we show (a) images with user-scribbles provided in [Gulshan *et al.*, 2010], (b) geodesic distance, (c) results of our method, and (d) geodesic segmentation [Bai and Sapiro, 2007]. The distance is visualized as likelihood forms. The results of method [Bai and Sapiro, 2007] are obtained by thresholding the distance maps.

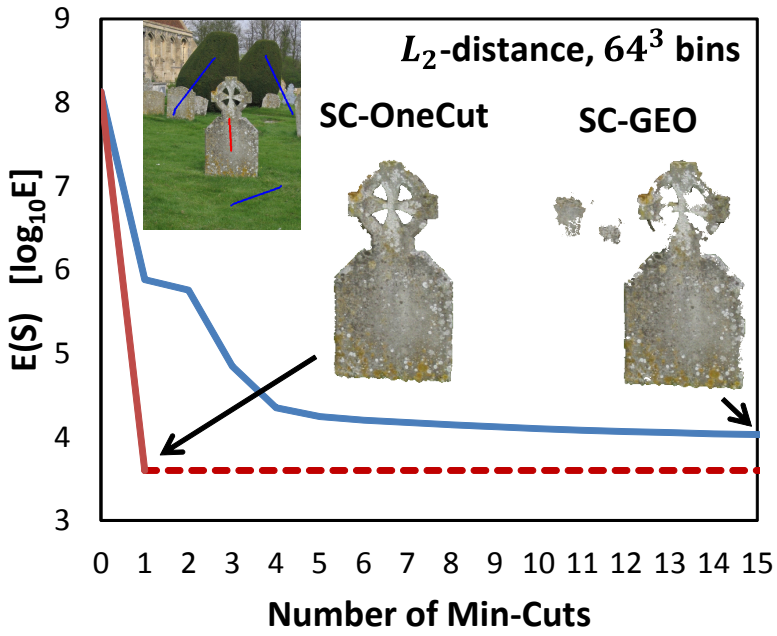


Figure 4.14 Energy convergence comparison of SC-OneCut and SC-GEO (1). We show the results of a difficult camouflage example "grave". The proposed **SC-OneCut** finds a good approximate solution in only one min-cut.

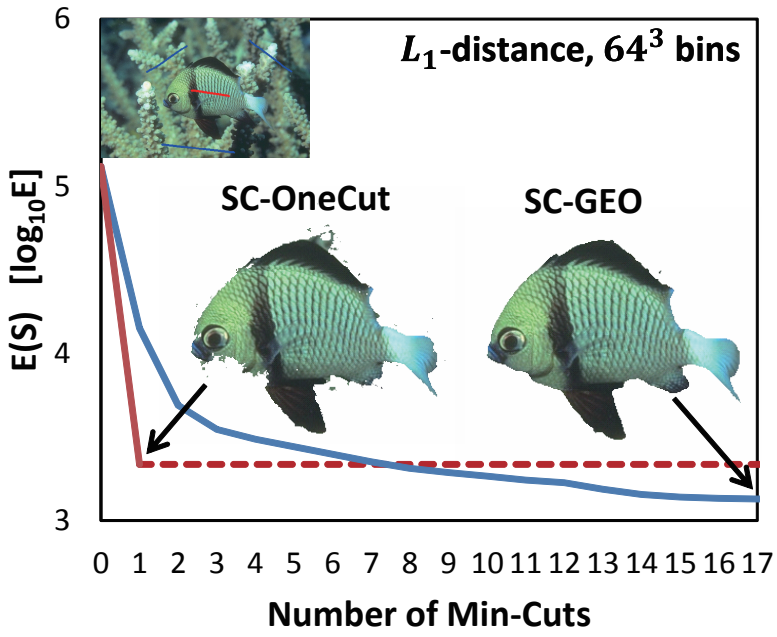


Figure 4.15 Energy convergence comparison of SC-OneCut and SC-GEO (2). We show the results of a difficult camouflage example "209070". The proposed **SC-OneCut** finds a good approximate solution in only one min-cut.

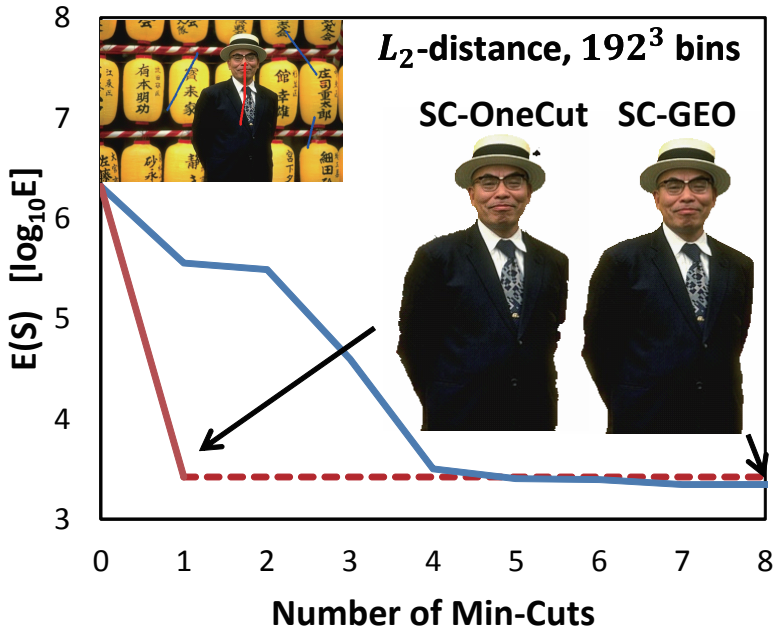


Figure 4.16 Energy convergence comparison of SC-OneCut and SC-GEO (3). We show the results of a difficult camouflage example "65019". The proposed **SC-OneCut** finds a good approximate solution in only one min-cut.

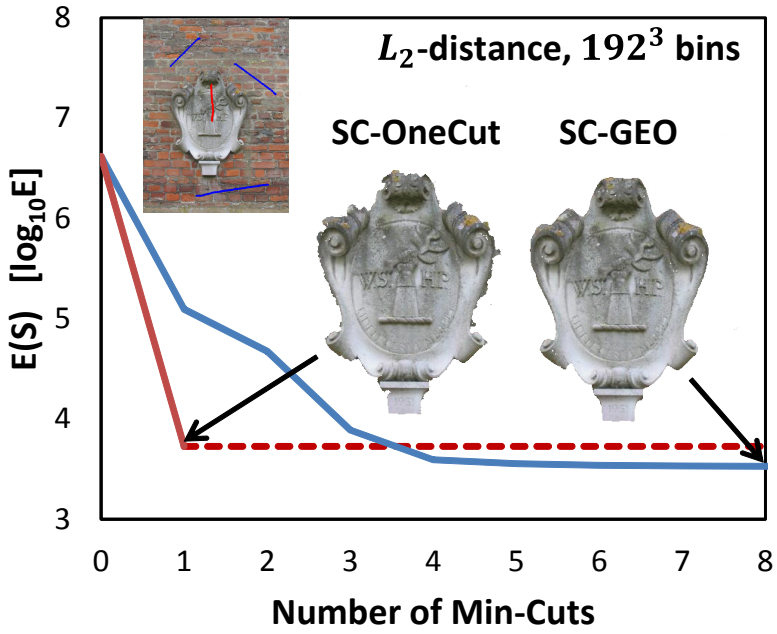


Figure 4.17 Energy convergence comparison of SC-OneCut and SC-GEO (4). We show the results of a difficult camouflage example "memorial". The proposed **SC-OneCut** finds a good approximate solution in only one min-cut.

4.8 Conclusions

4.8.1 Summary

In this work, we proposed an efficient optimization method that can be used for a variety fo non-linear higher-order terms. The proposed method was presented as an extension of SSP [Narasimhan and Bilmes, 2005] and AC [Ayed *et al.*, 2013], by pointing out their close connection. Unlike [Ayed *et al.*, 2013], our method allows bidirectional optimization and achieved greater accuracy and convergence than both methods. The bidirectionality was the key factor for the application to multiple-model segmentation problems. Also, we showed that the method can find good approximate solutions in only a single min-cut, given sparse user-scribbles are provided, which is a common situation in interactive segmentation. We hope that this work stimulate researchers in this field and promote the use of such non-linear terms, which are now shown to be efficiently optimized.

4.8.2 Future Directions

We discuss the current limitations and potential applications of our method, and present the future directions of this work.

Grouping Scheme

The thresholding scheme for making grouped permutations works well but somewhat ad-hoc. The value of the threshold τ should reflect the quality of the current segments, therefore an adaptive thresholding scheme based on the value of *e.g.* current energy function values $E(S^t)$ may be possible.

Other Move Making Methods

We used expansion and swap algorithms for multi-model distribution problems but we can use other generalized move making schemes such as α-expansion β-shrink moves [Schmidt and Alahari, 2011] and contraction moves [Woodford *et al.*, 2012], by which the method may achieve even greater convergence.

Video Segmentation

We believe that our bidirectional optimization is advantageous in video segmentation, because the object regions between neighboring frames are similar so we can use the results of the previous frames as the initial segmentations. But we leave this as our future work.

5 Conclusion

5.1 Summary of This Thesis

We studied energy minimization of continuous pairwise MRFs and discrete higher-order MRFs using GC.

For the first type of MRFs, we proposed an efficient inference method for accurate stereo vision. In spite of the huge solution value space, the proposed method efficiently finds good approximate solutions by incorporating spatial propagation techniques into GC based MRF optimization. The prosed method was designed for taking advantage of intrinsic properties of GC, and at the same time it was carefully tailored for the specifics of considered applications, *i.e.*, piecewise linearity of scenes. The proposed method was evaluated using the Middlebury stereo benchmark [Scharstein and Szeliski, 2001] and achieved the state-of-the-art performance among more than 150 stereo algorithms.

For the second type of MRFs, we proposed an efficient and general optimization method for various kinds of non-linear functions. The proposed method was presented as a generalization of two approaches [Ayed *et al.*, 2013; Narasimhan and Bilmes, 2005], and achieved about an order of magnitude greater accuracy than the current state-of-the-art method [Ayed *et al.*, 2013]. The proposed method was further applied to multiple distribution matching problems, which is the first work that addresses general multiple distribution matching problems. Furthermore, we showed that the proposed method can yield approximate solutions to higher-order MRF energies in only a single minimum cut, given sparse user-scribbles are provided.

5.2 Future Directions

Continuous MRF Optimization for Other Applications

We believe that our optimization strategy presented in Chapter 3 is not limited to the current binocular stereo matching problems but can be applied for more general corresponding field estimation such as multi-view stereo and optical flow. For multi-view stereo problems, the treatment of visibility and occlusion issues becomes more important. On the other hand, optical

flow is a problem of estimating dense motions between two frames of a video sequence, where each pixel is assigned a 2D label of horizontal and vertical disparities. When motions or displacements are small, classical continuous optimization approaches are effective. For large displacement optical flow problems, discrete optimization approaches have been gathering attention [Xu *et al.*, 2012; Lempitsky *et al.*, 2008], where non-convex energies are first optimized by discrete optimizers to be further refined by convex optimizers. By using our method, it may be possible to estimate highly complex, per-pixel affine flow models for achieving accurate optical flow.

Higher-Order Terms for Continuous MRFs

The use of higher-order terms is currently limited to discrete MRF formulations. However, we may be able to optimize continuous higher-order terms by extending our discrete optimization method, just similarly to the discrete-continuous optimization approach presented in Chapter 3. In matting problems, for example, a continuous alpha value is estimated for each pixel, which can be seen as a generalized problem of binary segmentation. Therefore, the use of higher-order terms is expected to be effective for such matting problems as well.

References

Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R. (2011). Building Rome in a Day. *Communications of the ACM*, **54**(10), 105–112.

Aherne, F. J., Thacker, N. A., and Rockett, P. (1998). The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data. *Kybernetika*, **34**(4), 363–368.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **33**(5), 898–916.

Ayed, I. B., Chen, H. M., Punithakumar, K., Ross, I., and Li, S. (2010). Graph cut segmentation with a global constraint: Recovering region distribution via a bound of the Bhattacharyya measure. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3288–3295.

Ayed, I. B., Gorelick, L., and Boykov, Y. (2013). Auxiliary Cuts for General Classes of Higher Order Functionals. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1304–1311.

Bai, X. and Sapiro, G. (2007). A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1–8.

Bai, X., Wang, J., Simons, D., and Sapiro, G. (2009). Video SnapCut: robust video object cutout using localized classifiers. *Proceedings of SIGGRAPH (ACM Transactions on Graphraphics)*, **28**(3), 70:1–70:11.

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *Proceedings of SIGGRAPH (ACM Transactions on Graphraphics)*, **28**(3), 24:1–24:11.

Barnes, C., Shechtman, E., Goldman, D. B., and Finkelstein, A. (2010). The Generalized PatchMatch Correspondence Algorithm. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 29–43.

Besse, F., Rother, C., Fitzgibbon, A., and Kautz, J. (2012). PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 132.1–132.11.

Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, **35**, 99–109.

Birchfield, S. and Tomasi, C. (1998). A Pixel Dissimilarity Measure That is Insensitive to Image Sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **20**(4), 401–406.

Bleyer, M., Rhemann, C., and Rother, C. (2011). PatchMatch Stereo - Stereo Matching with Slanted Support Windows. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 14.1–14.11.

Boros, E., Hammer, P. L., and Sun, X. (1991). Network Flows and Minimization of Quadratic Pseudo-Boolean Functions. Technical Report RRR 17-1991, RUTCOR Research Report.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Boykov, Y. and Jolly, M. P. (2001). Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 105–112.

Boykov, Y. and Kolmogorov, V. (2004). An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **26**(9), 1124–1137.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **23**(11), 1222–1239.

Boykov, Y., Komodakis, N., Kolmogorov, V., and Torr, P. (2007). Discrete Optimization in Computer Vision. In *Tutorials at International Conference on Computer Vision (ICCV)*. http://www.csd.uoc.gr/~komod/ICCV07_tutorial/.

Criminisi, A., Sharp, T., and Blake, A. (2008). Geos: Geodesic image segmentation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 99–112.

Delong, A., Osokin, A., Isack, H. N., and Boykov, Y. (2012). Fast Approximate Energy Minimization with Label Costs. *International Journal of Computer Vision (IJCV)*, **96**(1), 1–27.

Felzenszwalb, P. and Huttenlocher, D. (2004). Efficient Belief Propagation for Early Vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 261–268.

Freedman, D. and Zhang, T. (2005). Interactive graph cut based segmentation with shape priors. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 755–762.

Fujishige, S. (2005). *Submodular Functions and Optimization*, volume 58. Elsevier Science.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **6**(6), 721–741.

Gorelick, L., Schmidt, F. R., Boykov, Y., Delong, A., and Ward, A. D. (2012). Segmentation with non-linear regional constraints via line-search cuts. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 583–0597.

Gorelick, L., Schmidt, F. R., and Boykov, Y. (2013). Fast Trust Region for Segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1714–1721.

Gulshan, V., Rother, C., Criminisi, A., Blake, A., and Zisserman, A. (2010). Geodesic star convexity for interactive image segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3129–3136.

Hammer, P., Hansen, P., and Simeone, B. (1984). Roof Duality, Complementation and Persistency in Quadratic 0-1 Optimization. *Mathematicl Programming*, **28**, 121–155.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition.

He, K., Sun, J., and Tang, X. (2013). Guided Image Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **35**(6), 1397–1409.

Heise, P., Klose, S., Jensen, B., and Knoll, A. (2013). PM-Huber: PatchMatch with Huber Regularization for Stereo Matching. In *Proceedings of International Conference on Computer Vision (ICCV)*. (accepted).

Hong, L. and Chen, G. (2004). Segment-based Stereo Matching using Graph Cuts. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 74–81.

Hosni, A., Gelautz, M., and Bleyer, M. (2012). Accuracy-Efficiency Evaluation of Adaptive Support Weight Techniques for Local Stereo Matching. In *Proceedings of DAGM/OAGM Symposium*, pages 337–346.

Isack, H. and Boykov, Y. (2012). Energy-Based Geometric Multi-model Fitting. *International Journal of Computer Vision (IJCV)*, **97**(2), 123–147.

Iyer, R., Jegelka, S., and Bilmes, J. (2013). Fast Semidifferential-based Submodular Function Optimization. volume 28, pages 855–863.

Jebara, T., Kondor, R., and Howard, A. (2004). Probability Product Kernels. *Journal of Machine Learning Research*, **5**, 819–844.

Jegelka, S. and Bilmes, J. (2011). Submodularity beyond Submodular Energies: Coupling Edges in Graph Cuts. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1897–1904.

Kappes, J. H., Andres, B., Hamprecht, F. A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B. X., Lellmann, J., Komodakis, N., and Rother, C. (2013). A Comparative Study of Modern Inference Techniques for Discrete Energy Minimization Problems. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Klaus, A., Sormann, M., and Karner, K. (2006). Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, volume 3, pages 15–18.

Kohli, P., Osokin, A., and Jegelka, S. (2013). A Principled Deep Random Field Model for Image Segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1971–1978.

Kolmogorov, V. (2006). Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **28**(10), 1568–1583.

Kolmogorov, V. and Rother, C. (2007). Minimizing Nonsubmodular Functions with Graph Cuts – A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **29**(7), 1274–1279.

Kolmogorov, V. and Zabih, R. (2001). Computing Visual Correspondence with Occlusions using Graph Cuts. In *Proceedings of International Conference on Computer Vision (ICCV)*, volume 2, pages 508–515.

Kolmogorov, V. and Zabih, R. (2002). Multi-camera Scene Reconstruction via Graph Cuts. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 82–96.

Kolmogorov, V. and Zabin, R. (2004). What Energy Functions Can Be Minimized via Graph Cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **26**(2), 147–159.

Lempitsky, V., Rother, C., and Blake, A. (2007). LogCut - Efficient Graph Cut Optimization for Markov Random Fields. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1–8.

Lempitsky, V., Roth, S., and Rother, C. (2008). FusionFlow: Discrete-Continuous Optimization for Optical Flow Estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lempitsky, V., Kohli, P., Rother, C., and Sharp, T. (2009). Image segmentation with a bounding box prior. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 277–284.

Lempitsky, V., Rother, C., Roth, S., and Blake, A. (2010). Fusion Moves for Markov Random Field Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **32**(8), 1392–1405.

Liu, J., Sun, J., and Shum, H. Y. (2009). Paint selection. *Proceedings of SIGGRAPH (ACM Transactions on Graphraphics)*, **28**, 1–7.

Lu, J., Shi, K., Min, D., Lin, L., and Do, M. (2012). Cross-based Local Multipoint Filtering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 430–0437.

Lu, J., Yang, H., Min, D., and Do, M. N. (2013). Patch Match Filter: Efficient Edge-Aware Filtering Meets Randomized Search for Fast Correspondence Field Estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1854–1861.

Mizukami, Y., Okada, K., Nomura, A., Nakanishi, S., and Tadamura, K. (2012). Sub-Pixel Disparity Search for Binocular Stereo Vision. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 364–367.

Monasse, P. (2011). Quasi-Euclidean Epipolar Rectification. *Image Processing On Line*. http://www.ipol.im/pub/art/2011/m_qer/.

Narasimhan, M. and Bilmes, J. A. (2005). A Submodular-supermodular Procedure with Applications to Discriminative Structure Learning. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 404–412.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer series in operations research and financial engineering. Springer.

Olsson, C. and Boykov, Y. (2012). Curvature-based Regularization for Surface Approximation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1576–1583.

Olsson, C., Ulen, J., and Boykov, Y. (2013). In Defense of 3D-Label Stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1730–1737.

Pham, V.-Q., Takahashi, K., and Naemura, T. (2010). Real-Time video matting based on bilayer segmentation. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, volume 2, pages 489–501.

Pham, V.-Q., Takahashi, K., and Naemura, T. (2011). Foreground-Background Segmentation using Iterated Distribution Matching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2113–2120.

Price, B. L., Morse, B., and Cohen, S. (2010). Geodesic Graph Cut for Interactive Image Segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3288–3295.

Punithakumar, K., Yuan, J., Ayed, I. B., Li, S., and Boykov, Y. (2012). A Convex Max-Flow Approach to Distribution-Based Figure-Ground Separation. *SIAM Journal on Imaging Sciences*, **5**, 1333–1354.

Punithakumar, K., Yuan, J., Ayed, I. B., Boulanger, P., and Noga, M. (2013). A GPU Accelerated Convex Max-Flow Approach to Segmentation of 4-D Left-Ventricular Ultrasound. In *NVIDIA GPU Technology Conference*.

Rhemann, C., Hosni, A., Bleyer, M., Rother, C., and Gelautz, M. (2011). Fast Cost-Volume Filtering for Visual Correspondence and Beyond. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3017–3024.

Rother, C., Kolmogorv, V., and Blake, A. (2004). GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts. *Proceedings of SIGGRAPH (ACM Transactions on Graphraphics)*, **23**, 309–314.

Rother, C., Kolmogorov, V., Minka, T., and Blake, A. (2006). Cosegmentation of Image Pairs by Histogram Matching–Incorporating a Global Constraint into MRFs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 993–100.

Scharstein, D. and Szeliski, R. (2001). Middlebury Stereo Benchmark. http://vision.middlebury.edu/stereo/.

Schmidt, M. and Alahari, K. (2011). Generalized Fast Approximate Energy Minimization via Graph Cuts: a-Expansion b-Shrink Moves. In *Proceedings of Conference on Uncertainty in Artificial Intelligence*, pages 653–660.

Schrijver, A. (2000). A Combinatorial Algorithm Minimizing Submodular Functions in Strongly Polynomial Time. *Journal of Combinational Theory Ser. B*, **80**(2), 346–355.

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C. (2008). A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **30**(6), 1068–1080.

Taniai, T., Pham, V.-Q., Takahashi, K., and Naemura, T. (2012). Image Segmentation using Dual Distribution Matching. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 74.1–74.11.

Tao, H., Sawhney, H., and Kumar, R. (2001). A Global Matching Framework for Stereo Computation. In *Proceedings of International Conference on Computer Vision (ICCV)*, volume 1, pages 532–539.

Tomasi, C. and Manduchi, R. (1998). Bilateral Filtering for Gray and Color Images. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 839–846.

Wang, L. and Yang, R. (2011). Global Stereo Matching Leveraged by Pparse Ground Control Points. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3033–3040.

Wang, Z.-F. and Zheng, Z.-G. (2008). A Region based Stereo Matching Algorithm using Cooperative Optimization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wei, Y. and Quan, L. (2005). Asymmetrical Occlusion Handling Using Graph Cut for Multi-View Stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 902–909.

Woodford, O., Torr, P., Reid, I., and Fitzgibbon, A. (2009). Global Stereo Reconstruction under Second-Order Smoothness Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **31**(12), 2115–2128.

Woodford, O. J., Reid, I. D., Torr, P. H. S., and Fitzgibbon, A. W. (2007). On New View Synthesis using Multiview Stereo. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 1120–1129.

Woodford, O. J., Pham, M.-T., Maki, A., Gherardi, R., Perbet, F., and Stenger, B. (2012). Contraction moves for geometric model fitting. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 181–194.

Xu, L., Jia, J., and Matsushita, Y. (2012). Motion Detail Preserving Optical Flow Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **34**(9), 1744–1757.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2000). Generalized Belief Propagation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 689–695.

Yoon, K.-J. and Kweon, I.-S. (2005). Locally Adaptive Support-Weight Approach for Visual Correspondence Search. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 924–931.

Zhang, K., Lu, J., and Lafruit, G. (2009). Cross-Based Local Stereo Matching Using Orthogonal Integral Images. *IEEE Transactions on Circuits and Systems for Video Technology*, **19**(7), 1073–1079.

List of Publications

Journal Papers

[1] 谷合竜典, ファンヴェトクォク, 高橋桂太, 苗村健. (2013). “前景・背景色分布の同時マッチングによる画像セグメンテーション”, 電子情報通信学会論文誌 D, volume J96-D, number 8, pages 1764–1777.

International Conferences

[2] **Tatsunori Taniai**, Pham Viet-Quoc, Keita Takahashi, and Takeshi Naemura. (2012). “Image Segmentation using Dual Distribution Matching”. In *Proceedings of the 23rd British Machine Vision Conference (BMVC)*, pages 74.1–74.11.

[**accepted as oral – acceptance rate 8%**]

Domestic Conferences

[3] 谷合竜典, ファンヴェトクォク, 高橋桂太, 苗村健. (2012). “前景および背景色分布の同時マッチングによる画像セグメンテーション”, 第 15 回 画像の理解・認識シンポジウム 予稿集, OS14-02.

[口頭発表 – 採択率 **37%**]

[4] 谷合竜典, 苗村健. (2013). “画素位置を埋め込んだ双色分布マッチングによる画像セグメンテーション”, 第 16 回 画像の理解・認識シンポジウム 予稿集, SS3-31.

Papers under Review

TWO MORE PAPERS including the main contents of this dissertation are currently under review.

Appendix

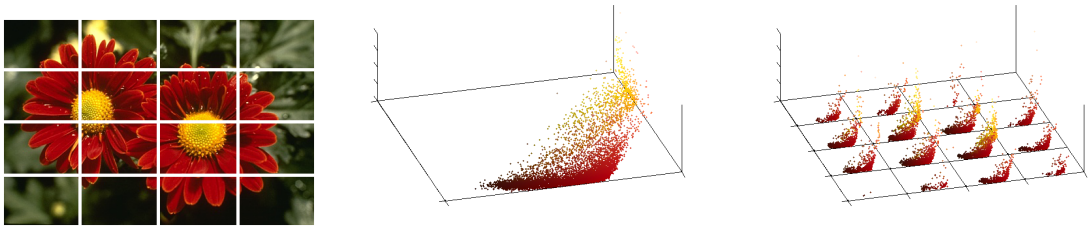
A

Incorporating Spatial Information into Distribution Matching Approaches

Abstract *We present an accurate image segmentation method that divides an image region into foreground and background regions. Our method is based on global distribution matching approaches, which seek segmentation by maximizing similarity between input color distributions and distributions computed from resulting segmented regions. Unlike previous distribution matching methods, we propose to additionally use pixel's coordinate information by augmenting pixel features from (R,G,B) to (R,G,B,X,Y) vectors, and we formulate our segmentation method as distribution matching using 5D histograms. To increase the robustness to the spatial kernel size of the histograms, we formulate our method as the weighted sum of multiple distribution matching terms with different spatial kernel sizes, where the weight of each matching term is adaptively estimated. The proposed technique can be used in a variety of existing distribution matching methods. In this paper the technique is combined with a recently proposed robust distribution matching method, by which the method yields even greater accuracy and robustness as we show in the experiments.*

A.1 Introduction

This paper addresses the problem of binary image segmentation when approximate color and spatial information of foreground and background regions are given as input. We can obtain such information *e.g.* from previous frames when successively processing video sequences. Such problems are often formulated as energy minimization of binary-labeling Markov random fields (MRFs) [Geman and Geman, 1984; Boykov and Jolly, 2001; Rother *et al.*, 2004; Liu *et al.*, 2009; Price *et al.*, 2010; Rother *et al.*, 2006; Ayed *et al.*, 2010; Pham *et al.*, 2011; Taniai *et al.*, 2012; Ayed *et al.*, 2013; Gorelick *et al.*, 2013]. In this approach the energy function is usually composed of two terms: the data term for measuring appearance consistencies between resulting segmentation and observed data (*e.g.* input color information of foreground and background), and the smoothness term for enforcing spatial smoothness on resulting segmentation.



(a) Spatial quantization of 4×4-grids (b) RGB distribution (c) RGBXY distribution

Figure A.1 The concept of 5D distributions. Pixel distributions of the foreground region of an image in (a) are shown in (b) and (c). Usually distribution matching methods use (b) RGB distributions with no spatial information. We propose to use (c) RGBXY distributions for exploiting spatial information in distribution matching approaches, where the spatial domain of RGBXY distributions is quantized, *e.g.*, using 4×4-grids shown in (a).

Using color similarity is the typical way to measure appearance consistencies, and such approaches can be categorized into two types: *local measures* and *global measures*. The local measures define likelihood of each pixel individually based on each pixel's color feature. This simple formulation is widely adopted [Boykov and Jolly, 2001; Rother *et al.*, 2004; Bai *et al.*, 2009; Liu *et al.*, 2009; Price *et al.*, 2010] because such energy functions can be optimally minimized by using graph cuts [Kolmogorov and Zabin, 2004; Boykov and Kolmogorov, 2004] if only pairwise smoothness terms are submodular. However, the optimal solutions of local measure methods often show visibly incorrect results, because they are strongly biased toward shorter boundary lengths (*a.k.a* shortcutting or shrinking bias) [Price *et al.*, 2010]. Recently the global measures have been gathering attention in computer vision [Rother *et al.*, 2006; Ayed *et al.*, 2010; Pham *et al.*, 2011; Taniai *et al.*, 2012; Ayed *et al.*, 2013; Gorelick *et al.*, 2013] because they have been shown to achieve greater accuracy and overcome the limitations of local measures. The global measures directly evaluate similarity between input distributions of pixel features and distributions computed from resulting segmented regions. Although this type of inference is NP-hard, it has been shown that approximate solutions can be efficiently estimated by iteratively applying graph cuts to some bound functions [Ayed *et al.*, 2010; Pham *et al.*, 2011; Taniai *et al.*, 2012; Ayed *et al.*, 2013] or to first-order approximation functions [Rother *et al.*, 2006; Gorelick *et al.*, 2013]. A recent work [Taniai *et al.*, 2012] has shown that the robustness of global measures can be significantly increased via dual distribution matching that simultaneously enforces consistencies between resulting segmentation and two (foreground and background) input distributions.

The use of spatial information, on the other hand, is widely studied in the context of local measure methods. For example, user's scribbles are used as hard constrains in interactive segmentation [Boykov and Jolly, 2001] and, furthermore, likelihood is spatially propagated from the scribbles in geodesic segmentation [Price *et al.*, 2010]. In video segmentation, motions are used in [Pham *et al.*, 2010; Bai *et al.*, 2009]. Although these works suggest the effectiveness of the use of spatial information for segmentation, the distribution matching approaches seem so far not

aware of spatial information.

In this paper, we develop a method for exploiting pixel's spatial information in global distribution matching approaches. The key idea of our method is to augment pixel features from (R,G,B) to 5D vectors of (R,G,B,X,Y), and we formulate our method as distribution matching using 5D histograms (see Figure A.1). To increase the robustness to the spatial kernel size of histograms, our method is formulated as the weighted sum of multiple distribution matching terms using different kernel sizes with adaptively estimated weights. The use of this formulation has several benefits: it can be easily combined with previously proposed distribution matching methods [Rother *et al.*, 2006; Ayed *et al.*, 2010; Pham *et al.*, 2011; Taniai *et al.*, 2012; Ayed *et al.*, 2013; Gorelick *et al.*, 2013]. Particularly we use a recently proposed dual distribution matching method [Taniai *et al.*, 2012] as our baseline, which seems much more robust and accurate than single distribution matching methods [Rother *et al.*, 2006; Ayed *et al.*, 2010; Pham *et al.*, 2011; Ayed *et al.*, 2013; Gorelick *et al.*, 2013]; it uses input spatial information as *soft constraints*, by which the method is robust to large motions or dynamic scene changes in video segmentation; it does not require motion tracking or optical flow, which is often computationally expensive.

A.2 Related Works

Spatial information is actively exploited in interactive segmentation, where a user specifies segmentation clues by scribbling foreground and background regions or giving a bounding box to the foreground region.

A typical example that uses user's scribbles is the interactive graph cuts proposed by Boykov and Jolly [Boykov and Jolly, 2001]. The method first learns color distributions of foreground and background from scribbled pixels and imposes hard constraints on those pixels as *seeds*. This is also the pioneer work that uses graph cuts for image segmentation. A similar method is proposed in [Liu *et al.*, 2009]. Recently Price *et al.* [Price *et al.*, 2010] proposes to use the geodesic distance in interactive segmentation, where likelihood of being foreground and background is spatially propagated from hard-constrained scribbled pixels using a geodesic metric.

GrabCut by Rother *et al.* [Rother *et al.*, 2004] proposes to use a bounding box for interactive segmentation. In this method, initial color distributions of foreground and background are learned from the inside and outside of the bounding box using GMMs, and the outside pixels are hard labeled as background. Lempitsky *et al.* [Lempitsky *et al.*, 2009] proposes a bounding box prior which exploits an assumption that the foreground region is tightly enclosed by the bounding box.

All of these methods described above require user's inputs and treat them as hard constraints. However, the use of hard constraints is sometimes disadvantageous. For example when processing a video sequence, using the spatial information of objects learned from the previous frames as hard constraints may cause inevitable errors due to large motions or dynamic scene changes. Our method uses the spatial information rather as *soft constraints*.

In the context of video segmentation, shape priors [Freedman and Zhang, 2005] are often

used, which enforce segmentation boundaries happen near the boundaries of a shape template (*e.g.* the segmentation results of previous frames) by setting higher smoothness costs *w.r.t.* the distance from the template's boundaries. As we will see in the experiments, this method is less robust to dynamic scene changes compared with our method. Bai *et al.* [Bai *et al.*, 2009] proposes localized classifiers, which perform graph-cut segmentation in small overlapping windows along pre-estimated boundaries, where color distributions are learned in each local window. The idea of the localized color models is similar to our RGBXY histograms in spirit. However, the method [Bai *et al.*, 2009] requires relatively accurate initial segmentation. In fact, it uses SIFT feature matching and optical flow for propagating the segmentation results of the previous frames in order to obtain accurate initial segmentation.

To summarize, the use of spatial information is popular in local measure methods, particularly in interactive segmentation, but much less used in global measure approaches. Since our method takes advantage of good performance of global measures, it performs quite well without using any hard constrained pixels or expensive motion tracking.

A.3 Proposed Method

A.3.1 Review of Dual Distribution Matching

Before presenting our method, we briefly review a dual distribution matching method (or DDM) proposed in [Taniai *et al.*, 2012], which we use as our baseline method. Let us consider a problem of finding a binary-labeling function $\mathcal{L}$ that assigns the foreground F or background B label to the set of all pixels $P \subset \mathbb{Z}^2$ as $\mathcal{L}_p = \mathcal{L}(p) : P \to \{F, B\}$. Let $\mathcal{P}_l^{\mathcal{L}} : Z \to [0, 1]$ $(l = F, B)$ be the distribution (histogram) of pixel features $I_p \in Z$ computed from the foreground $(l = F)$ or background $(l = B)$ region of the segmentation $\mathcal{L}$. The distribution $\mathcal{P}_l^{\mathcal{L}}$ is normalized so that it sums up to 1.

We assume approximate distributions of foreground and background regions $\mathcal{H}_F \simeq \mathcal{P}_F^{\mathcal{L}^*}$ and $\mathcal{H}_B \simeq \mathcal{P}_B^{\mathcal{L}^*}$ are given as input. Using only this information, we estimate the true segmentation $\mathcal{L}^*$ by minimizing the following energy function:

$$\mathcal{E}(\mathcal{L}|\mathcal{H}_F,\mathcal{H}_B) = \underbrace{\lambda_F \mathcal{M}_F(\mathcal{L}|\mathcal{H}_F)}_{\text{foreground matching}} + \underbrace{\lambda_B \mathcal{M}_B(\mathcal{L}|\mathcal{H}_B)}_{\text{background matching}} + \underbrace{\mathcal{S}(\mathcal{L})}_{\text{smoothness}} \tag{A.1}$$

Here, the matching terms $\mathcal{M}_l(\mathcal{L}|\mathcal{H}_l)$ are formulated as the dissimilarity between the resulting and input distributions:

$$\mathcal{M}_l(\mathcal{L}|\mathcal{H}_l) = -\mathcal{B}\left(\mathcal{P}_l^{\mathcal{L}}, \mathcal{H}_l\right) \quad (l = F, B) \tag{A.2}$$

and we use the Bhattacharyya coefficient [Bhattacharyya, 1943; Aherne *et al.*, 1998] for the

similarity measure $\mathcal{B}(,)$ that is defined and bounded as

$$\mathcal{B}(f,g) = \sum_{z \in Z} \sqrt{f(z)g(z)} \ \in [0,1]. \tag{A.3}$$

$\mathcal{S}(\mathcal{L})$ is the smoothness term defined as

$$\mathcal{S}(\mathcal{L}) = \lambda \sum_{(p,q)\in\mathcal{N}} \frac{\delta_{\mathcal{L}_p \neq \mathcal{L}_q}}{|p-q|} \left(\frac{1}{1+|I_p - I_q|^2} + \epsilon \right). \tag{A.4}$$

Here, $\mathcal{N}$ is the set of 8-neighbor pixel pairs, $\delta_{true} = 1$, $\delta_{false} = 0$, $I_p \in \mathbb{R}^3$ is a color vector of a pixel p, and λ and ϵ are user-defined parameters. The physical meaning of this energy function $E(\mathcal{L}|\mathcal{H}_F, \mathcal{H}_B)$ is to simultaneously enforce the resulting distributions $\mathcal{P}_F^{\mathcal{L}}$ and $\mathcal{P}_B^{\mathcal{L}}$ get close to input distributions $\mathcal{H}_F$ and $\mathcal{H}_B$, respectively.

It has been shown in [Taniai *et al.*, 2012] that the optimal weights of the two matching terms λ_F and λ_B can be estimated as

$$\lambda_F = \sqrt{\eta_F \, r_F^{\mathcal{L}}}, \ \ \lambda_B = \sqrt{\eta_B \, r_B^{\mathcal{L}}}. \tag{A.5}$$

Here, $\eta_F \in [0,1]$ ($\eta_B = 1 - \eta_F$) is the rate of the foreground (background) region size *w.r.t.* the entire image size estimated via matching entire image distributions as

$$\eta_F = \text{argmin}\, \mathcal{M}_\Omega(\eta|\mathcal{H}_F, \mathcal{H}_B) \tag{A.6}$$

where

$$\mathcal{M}_\Omega(\eta|\mathcal{H}_F, \mathcal{H}_B) = -\mathcal{B}\left(\mathcal{H}_\Omega, \tilde{\mathcal{H}}_\Omega(\eta)\right). \tag{A.7}$$

Here, $\mathcal{H}_\Omega$ is the distribution of the entire image region, and $\tilde{\mathcal{H}}_\Omega$ is its approximation by the sum of input distributions $\tilde{\mathcal{H}}_\Omega(\eta) = \eta\,\mathcal{H}_F + (1-\eta)\,\mathcal{H}_B$. The minimization of Equation (A.6) can be easily obtained because the function $\mathcal{M}_\Omega(\eta|\mathcal{H}_F, \mathcal{H}_B)$ is convex and the only variable η is limited in $[0,1]$. The value $r_F^{\mathcal{L}}$ (or $r_B^{\mathcal{L}}$) in Equation (A.5) is the rate of foreground (background) region size *w.r.t.* the entire image size computed from segmentation $\mathcal{L}$. An intuitive interpretation of the weights of Equation (A.5) is that each matching term $\mathcal{M}_l(\mathcal{L}|\mathcal{H}_l)$ ($l = F, B$) is weighted by the geometric-mean of region size rates computed by the two ways (*i.e.* from the input distributions or from the output segmentation).

The minimum solution of $E(\mathcal{L}|\mathcal{H}_F, \mathcal{H}_B)$ is estimated by iteratively minimizing the upper-bound functions of $E(\mathcal{L}|\mathcal{H}_F, \mathcal{H}_B)$ using graph cuts. Please refer to [Taniai *et al.*, 2012] for the detailed optimization procedure.

A.3.2 Formulation

We present our segmentation method by borrowing the framework of a robust distribution matching method, DDM [Taniai *et al.*, 2012]. Unlike DDM [Taniai *et al.*, 2012] and other previous distribution matching methods that use 3D vectors of (R,G,B) as pixel features, we use 5D vectors of (R,G,B,X,Y) for taking advantage of spatial information. Specifically, the pixel feature space Z is extended from the RGB color space $[0, 255]^3$ to the RGB-XY space $([0, 255]^3 \times [0, \text{width} - 1] \times [0, \text{height} - 1])$. To robustly estimate the distributions from sample pixels, we use histograms $\mathcal{P}^{\mathcal{L}}_{l|\tilde{Z}} : \tilde{Z} \to [0, 1]$ computed by a quantized feature space $\tilde{Z}$ where the RGB-values are quantized uniformly by some bin-width, and XY-values are quantized by $N \times N$-grids. For example, when RGB-values are quantized by a bin-width of 4, and XY-values are quantized by 4×4-grids, the histogram $\mathcal{P}^{\mathcal{L}}_{l|\tilde{Z}}$ consists of $64^3 \times 4^2$ bins. The concept of this 5D histogram is illustrated in Figure A.1.

In this approach the quantization level of XY-values directly influences the performance of the segmentation method; it may become too sensitive to objects' positions with fine-quantization, and becomes in turn totally insensitive with 1×1-grid quantization. To deal with this issue, we re-formulate the matching terms $\mathcal{M}_l(\mathcal{L}|\mathcal{H}_l)$ of Equation (A.2) as the weighted sum of multiple matching terms with different quantization levels $\tilde{Z} \in Q$:

$$\mathcal{M}_l(\mathcal{L}|\mathcal{H}_l) = -\sum_{\tilde{Z}\in Q} \omega_{\tilde{Z}}\, \mathcal{B}\left(\mathcal{P}^{\mathcal{L}}_{l|\tilde{Z}}, \mathcal{H}_{l|\tilde{Z}}\right) \quad (l = F, B). \tag{A.8}$$

Here, the weight of each quantization level $\omega_{\tilde{Z}}$ is normalized so that $\sum_{\tilde{Z}\in Q} \omega_{\tilde{Z}} = 1$. The definition of the weights $\omega_{\tilde{Z}}$ is discussed in the next section. Our proposed method is to use the energy function $E(\mathcal{L}|\mathcal{H}_F, \mathcal{H}_B)$ of DDM in Equation (A.1) but replace the definition of the matching terms $\mathcal{M}_l(\mathcal{L}|\mathcal{H}_l)$ with Equation (A.8). Note that if only one level of quantization $Q = \{\tilde{Z}\}$ is used and its spatial quantization is defined as 1×1-grid (*i.e.* use no spatial information), our method is equivalent to DDM [Taniai *et al.*, 2012]. By simultaneously evaluating consistencies with different quantization levels, we expect an increased robustness to the settings of the spatial kernel size of histograms. We shall denote our method using DDM by 5D-DDM.

Note that our new energy function can be optimized in the same manner with DDM [Taniai *et al.*, 2012] because the formulations are essentially equivalent. The only modifications we need are to make 5D histograms instead of 3D, and to sum the multiple matching terms with different quantization levels for calculating the energy function and cost values during the optimization.

A.3.3 Weights of Matching Terms with Different Quantization Levels

The simplest strategy for determining the weights of quantization levels $\omega_{\tilde{Z}}$ is to set them evenly as $\omega_{\tilde{Z}} = 1/|Q|$. Although this simple strategy yields good performance as will be shown in the experiments, we investigate a more sophisticated way specialized for DDM where we adaptively estimate the weights based on the accuracies of input distributions.

It has been shown in [Taniai *et al.*, 2012] that the dual distribution matching term $\mathcal{D}(\mathcal{L}|\mathcal{H}_F, \mathcal{H}_B) := \lambda_F \mathcal{M}_F(\mathcal{L}|\mathcal{H}_F) + \lambda_B \mathcal{M}_B(\mathcal{L}|\mathcal{H}_B)$ in Equation (A.1) with the estimated weights λ_F and λ_B of Equation (A.5) is lower-bounded by the minimum value of $\mathcal{M}_\Omega(\eta|\mathcal{H}_F, \mathcal{H}_B)$ in Equation (A.7), *i.e.*,

$$-1 \leq \mathcal{M}_\Omega(\eta_F|\mathcal{H}_F, \mathcal{H}_B) \leq \mathcal{D}(\mathcal{L}|\mathcal{H}_F, \mathcal{H}_B) \tag{A.9}$$

holds for an arbitrary $\mathcal{L}$. Furthermore, $\mathcal{M}_\Omega(\eta_F|\mathcal{H}_F, \mathcal{H}_B)$ takes its lowest bound of -1 when the input distributions $\mathcal{H}_F$ and $\mathcal{H}_B$ are given as ground truth. These facts suggest that the minimum matching value $\mathcal{M}_\Omega(\eta_F|\mathcal{H}_F, \mathcal{H}_B)$ is somehow related to the performance limitation of using $\mathcal{D}(\mathcal{L}|\mathcal{H}_F, \mathcal{H}_B)$ with given input distributions $\mathcal{H}_F$ and $\mathcal{H}_B$; when $\mathcal{M}_\Omega(\eta_F|\mathcal{H}_F, \mathcal{H}_B)$ is close to -1, using $\mathcal{D}(\mathcal{L}|\mathcal{H}_F, \mathcal{H}_B)$ is more realistic, and when close to 0 unrealistic. Based on this assumption we determine the weights of quantization levels $\omega_{\tilde{Z}}$ as below.

We first compute the matching value $\mathcal{M}_\Omega(\eta_F|\mathcal{H}_F, \mathcal{H}_B)$ using each quantization level $\tilde{Z} \in Q$. Let $D(\tilde{Z})$ denote such matching values defined as the Bhattacharyya distance as

$$D(\tilde{Z}) = -\log \mathcal{B}\left(\mathcal{H}_{\Omega|\tilde{Z}}, \tilde{\mathcal{H}}_{\Omega|\tilde{Z}}(\eta_F)\right) \geq 0. \tag{A.10}$$

Then we compute relative matching values $\bar{D}(\tilde{Z})$ by

$$\bar{D}(\tilde{Z}) = D(\tilde{Z}) - \min_{Z \in Q} D(Z). \tag{A.11}$$

As we prefer a bigger weight $\omega_{\tilde{Z}}$ for a smaller distance $\bar{D}(\tilde{Z})$, we define $\omega_{\tilde{Z}}$ using the Gaussian probability as

$$\omega_{\tilde{Z}} = \exp\left(-\bar{D}(\tilde{Z})/2\sigma^2\right) / \sum_{\tilde{Z} \in Q} \omega_{\tilde{Z}} \tag{A.12}$$

where σ is a user-defined parameter.

A.4 Experiments

In the following sections we assess the performance of our method in image and video segmentation. Throughout the experiments we use a laptop computer with a mobile version of Core i7 CPU (2.80 GHz) and 8 GB RAM. We implement our method using C++ and a graph cut implementation of [Boykov and Kolmogorov, 2004].

A.4.1 Image segmentation

In this section we evaluate the performance of our segmentation method using 50 images in GrabCut segmentation dataset [Rother *et al.*, 2004]. We compare the following six settings for our 5D-DDM method: (a) three levels of spatial quantization of 1×1, 2×2, and 4×4-grids with estimated weights $\omega_{\tilde{Z}}$ using $\sigma = 0.2$, (b) the same settings with (a) but use uniform weights for

$\omega_{\tilde{Z}}$, (c) two levels of spatial quantization of 1×1 and 2×2-grids, (d) 1×1-grid quantization (*i.e.* equivalent to DDM [Taniai *et al.*, 2012]) as the baseline method, (e) 2×2-grid quantization, and (f) 4×4-grid quantization. For (a)–(f), we use $\{\lambda, \epsilon\} = \{10^{-3}, 8\times10^{-4}\}$ and 64^3 quantization (bin-width 4) for the RGB color space as specified in [Taniai *et al.*, 2012]. As a reference we also compare with (g) interactive graph cuts [Boykov and Jolly, 2001] as a local measure method.

As the inputs, we make approximate pixel feature distributions of foreground and background regions ($\mathcal{H}_F$ and $\mathcal{H}_B$) from trimaps given by the dataset. Note that we use trimaps only for making input distributions and we do not use them as hard constraints. We use two accuracy measures, error pixel rate (EPR) and error to object ratio (EOR), defined as

$$\text{EPR} = \frac{\text{\# error pixels}}{\text{\# all pixels}}, \quad \text{EOR} = \frac{\text{\# error pixels}}{\text{\# true foreground pixels}}.$$

We show in Table A.1 the average EPRs, EORs and running times of the seven methods. Among them, (a) 5D-DDM using three levels of spatial quantization with estimated weights outperforms the other methods in both EPR and EOR evaluations. Comparing (a) with (b), our adaptive weight estimation allows a small improvement with almost no extra computational cost. Seeing the results of (d)–(f) that use a single quantization level, (e) 2×2-grid quantization improves the accuracy over (d) the baseline, but degrades when using (f) 4×4-grids. The combinational use of these three quantization levels improves the performance as shown in (b) and (c). Interestingly, despite that (f) 4×4-grid quantization alone degrades the performance, it yields improvements when combined with the other quantization levels as shown in (b) and (c).

Figure A.2 shows example results of three methods (a) 5D-DDM using three levels of quantization with estimated weights, (d) DDM, and (g) interactive graph cuts. Being a global measure method, (a) and (d) well preserve thin structures compared with (g) a local measure method. Also, (a) our method performs better than (b) DDM when foreground and background regions share similar colors, *e.g.*, see the results of *sheep*, where the sheep in the background is correctly labeled as background in (a).

A.4.2 Video Segmentation

We further evaluate the performance of our method by applying to a video sequence *foreman*, which consists of 100 frames with 352×288-size. We first manually segment the first frame, and apply our method for the rest of the frames using input distributions $\mathcal{H}_F$ and $\mathcal{H}_B$ learned from the result of the previous frames.

In Figure A.3 we show example frames of the segmentation results obtained by (a) 5D-DDM using three levels of spatial quantization, (d) DDM [Taniai *et al.*, 2012], and (d+) DDM [Taniai *et al.*, 2012] with shape priors [Freedman and Zhang, 2005] using previous frame's results as shape templates. Because the colors of the helmet is similar to the background colors, it is difficult even for (d) DDM to correctly separate foreground from background regions. By using spatial information (a) our method performs better than (d) DDM. In addition, our method seems more

Table A.1 Comparison of segmentation accuracies for 50 images of GrabCut dataset [Rother *et al.*, 2004]. The results of our methods, DDM Taniai *et al.* [2012], and BJGC Boykov and Jolly [2001] are shown.

	# grids of $\tilde{Z} \in Q$	EPR (mean±std)	EOR (mean±std)	time/image [sec]
(a) 5D-DDM (adaptive $\omega_{\tilde{Z}}$)	1^2, 2^2, 4^2	**1.07** ± **0.69** %	**6.38** ± **4.81** %	10.09
(b) 5D-DDM	1^2, 2^2, 4^2	1.09 ± 0.73 %	6.44 ± 4.97 %	10.04
(c) 5D-DDM	1^2, 2^2	1.13 ± 0.71 %	6.81 ± 5.14 %	4.49
(d) 5D-DDM (baseline; DDM)	1^2	1.23 ± 0.79 %	7.88 ± 6.55 %	2.32
(e) 5D-DDM	2^2	1.12 ± 0.74 %	6.80 ± 5.42 %	3.64
(f) 5D-DDM	4^2	1.35 ± 0.95 %	8.36 ± 7.88 %	7.56
(g) BJGC (local measures)	-	1.53 ± 0.96 %	10.4 ± 9.40 %	0.23

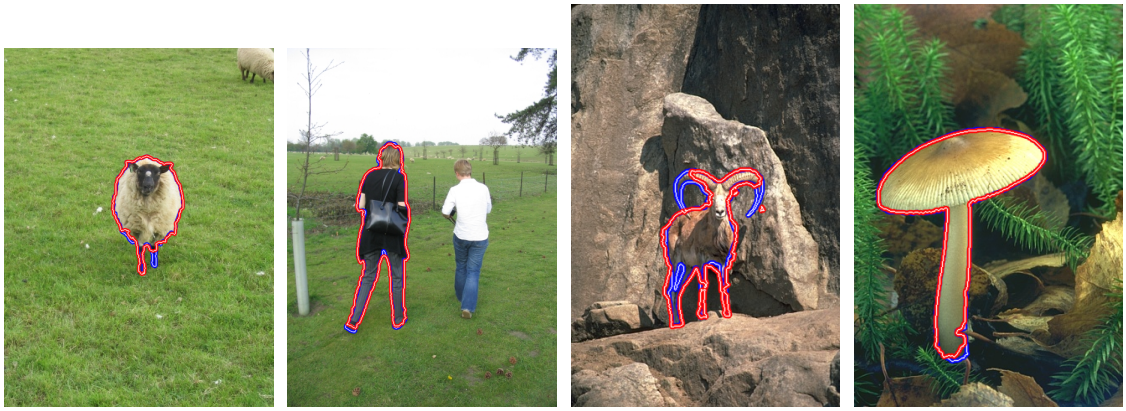
robust to dynamic scene changes than shape priors [Freedman and Zhang, 2005] used in (d+). For example, a hand that suddenly appears at the frame #89 is correctly labeled as foreground even using spatial information learned from the frame #88 where the hand is not shown yet. This is because the use of RGBXY histograms helps to resolve ambiguity in RGB-histogram matching, rather than directly constrains object regions as done by shape priors.

A.5 Conclusions

In this paper we present a method for incorporating pixel's spatial information into distribution matching approaches in the context of image segmentation. Our method is simply formulated as distribution matching using 5D histograms of augmented pixel vectors (R,G,B,X,Y), thus it can be used in various distribution matching methods. Particularly in this paper, our method is demonstrated using a recently proposed robust distribution matching method, dual distribution matching [Taniai *et al.*, 2012]. We show in the experiments that, by combining multiple distribution matching terms with various spatial kernel sizes of histograms, the accuracy and robustness of the baseline method can be further improved. Since spatial information is used as soft constraints our method is robust to dynamic scene changes in video segmentation.

Currently our method is demonstrated only using dual distribution matching [Taniai *et al.*, 2012] but can be used with other distribution matching methods [Rother *et al.*, 2006; Ayed *et al.*, 2010; Pham *et al.*, 2011; Ayed *et al.*, 2013; Gorelick *et al.*, 2013]. We leave the performance evaluations of those cases as our future work. Also, although the current implementation of our method is relatively slow, recent works by Punithakumar *et al.* [Punithakumar *et al.*, 2013, 2012] show that distribution matching methods can be efficiently performed in a parallel manner on GPUs using convex maxflow approaches. The development of an efficient GPU implementation of our method is thus another future direction.

sheep *person5* *304074* *208001*

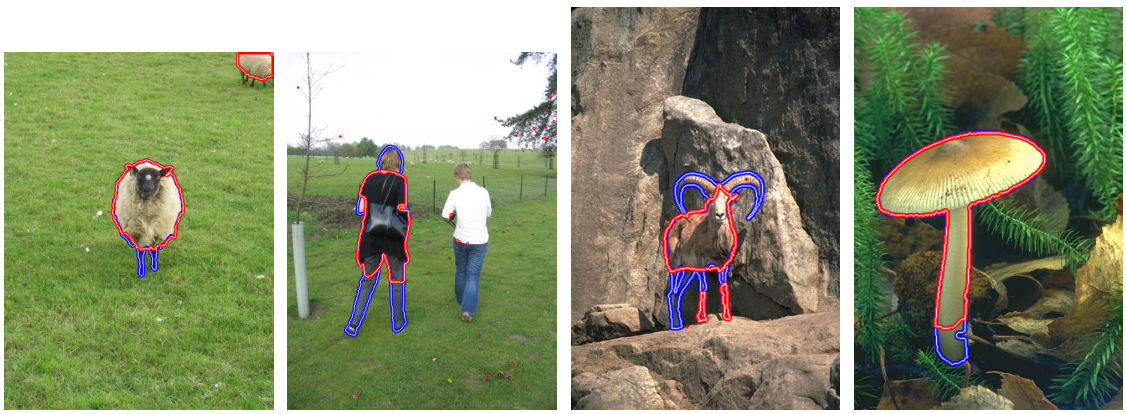


EPR: **0.267%**, EOR: **5.138%** EPR: **0.329%**, EOR: **4.866%** EPR: **1.290%**, EOR: **20.87%** EPR: **0.436%**, EOR: **3.398%**

(a) Proposed 5D-DDM

EPR: 1.086%, EOR: 20.89% EPR: 0.653%, EOR: 8.324% EPR: 1.701%, EOR: 27.53% EPR: 1.074%, EOR: 8.371%

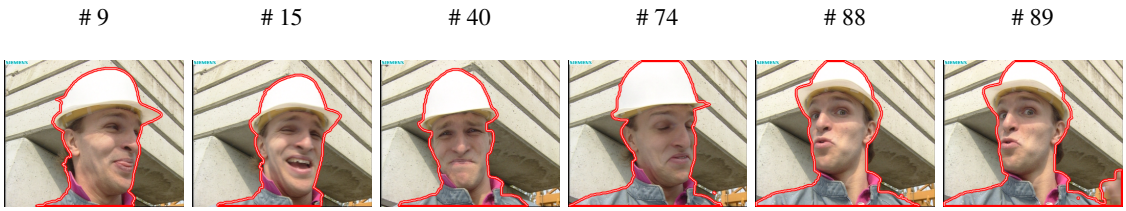
(d) DDM [Taniai *et al.*, 2012]



EPR: 1.198%, EOR: 24.83% EPR: 2.224%, EOR: 32.91% EPR: 2.333%, EOR: 37.74% EPR: 1.198%, EOR: 9.336%

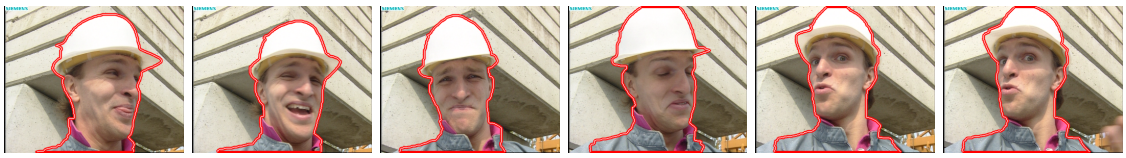
(g) BJGC [Boykov and Jolly, 2001]

Figure A.2 Segmentation results for GrabCut dataset [Rother *et al.*, 2004] using approximate input distributions learned from trimaps. We show example results of (a) the proposed 5D-DDM using three levels of spatial quantization of 1×1, 2×2, and 4×4-grids, (d) DDM [Taniai *et al.*, 2012] using no spatial information as the baseline method, and (g) interactive graph cuts [Boykov and Jolly, 2001] as a local measure method. Blue lines indicate the ground truth boundaries and red the resulting segmentation boundaries.



(a) Proposed 5D-DDM

(d) DDM [Taniai *et al.*, 2012]



(d+) DDM [Taniai *et al.*, 2012] with shape priors [Freedman and Zhang, 2005]

Figure A.3 Video segmentation results for "foreman". See also the supplementary video. Using spatial information (a) the proposed 5D-DDM performs better than (d) the baseline method. Because our method uses spatial information as soft constraints, it is robust to dynamic scene changes, *e.g.*, a hand that suddenly appears at the frame #89 is correctly labeled even using spatial information of the previous frame #88. Shape priors [Freedman and Zhang, 2005] used in (d+) seem less robust than our method.