Computing Visual Correspondence with Occlusions via Graph Cuts

Vladimir Kolmogorov   Ramin Zabih

vnk@cs.cornell.edu   rdz@cs.cornell.edu

Computer Science Department
Cornell University
Ithaca, NY 14853

Abstract

Several new algorithms for visual correspondence based on graph cuts [6, 13, 16] have recently been developed. While these methods give very strong results in practice, they do not handle occlusions properly. Specifically, they treat the two input images asymmetrically, and they do not ensure that a pixel corresponds to at most one pixel in the other image. In this paper, we present two new methods which properly address occlusions, while preserving the advantages of graph cut algorithms. We give experimental results for stereo as well as motion, which demonstrate that our methods perform well both at detecting occlusions and computing disparities.

1 Introduction

In the last few years, a new class of algorithms for visual correspondence has been developed that are based on graph cuts [6, 13, 16]. These methods give very strong experimental results; for example, a recent comparative study [17] of stereo algorithms found that one such algorithm gave the best results, with approximately 4 times fewer errors than standard methods such as normalized correlation. Unfortunately, existing graph cut algorithms do not treat occlusions correctly. In this paper, we present two new graph cut algorithms that handle occlusions properly, while maintaining the key advantages of graph cuts.

Occlusions are a major challenge for the accurate computation of visual correspondence. Occluded pixels are visible in only one image, so there is no corresponding pixel in the other image. For many applications, it is particularly important to obtain good results at discontinuities, which are places where occlusions often occur. Ideally, a pixel in one image should correspond to at most one pixel in the other image, and a pixel that correspond to no pixel in the other image should be labeled as occluded. We will refer to this requirement as *uniqueness*.

Most algorithms for visual correspondence do not enforce uniqueness. (We will discuss algorithms that enforce uniqueness when we summarize related work in section 4.) It is common to compute a disparity for each pixel in one (preferred) image. This treats the two images asymmetrically, and does not make full use of the information in both images. The recent algorithms based on graph cuts [6, 13, 16] are typical in this regard, despite their strong performance in practice.

The new algorithms proposed in this paper are based on energy minimization. Our methods are most closely related to the algorithms of [7], which can find a strong local minimum of a natural class of energy functions. We address the correspondence problem by constructing a problem representation and an energy function, such that a solution which violates uniqueness will have infinite energy. Constructing an appropriate energy function is non-trivial; for example, there are natural energy functions where it is NP-hard to even compute a local minimum. We consider two different energy functions, and show how to use graph cuts to compute a strong local minimum.

This paper begins with a discusion of the algorithms of [7]. We then give an overview of our algorithms, in which we discuss our problem representation and our choice of energy functions, and show how they enforce uniqueness. In section 4 we survey some related work, focusing on other algorithms that guarantee uniqueness. In sections 5 and 6 we show how to compute a local minimum of our energy functions in a strong sense using graph cuts. Experimental results are given in section 7.

2 Expansion moves and swap moves

Let $\mathcal{L}$ be the set of pixels in the left image, let $\mathcal{R}$ be the pixels in the right image, and let $\mathcal{P}$ be the set of all pixels: $\mathcal{P} = \mathcal{L} \cup \mathcal{R}$. The pixel p will have coordinates (p_x, p_y). In the classical approach to stereo, the goal is to compute, for each pixel in the *left* image, a label f_p which denotes a disparity value for a pixel p. The energy minimized in [7] is the Potts energy[1] of [15]

$$E(f) = \sum_{p \in \mathcal{L}} D_p(f_p) + \sum_{p,q \in \mathcal{N}} V_{p,q} \cdot T(f_p \neq f_q). \tag{1}$$

Here $D_p(f_p)$ is a penalty for the pixel p to have the disparity f_p, $\mathcal{N}$ is a neighborhood system for the pixels of the left image and $T(\cdot)$ is 1 if its argument is true and 0 otherwise. Minimizing this energy is NP-hard, so [7] gives two approximation algorithms. They involve the notion of moves.

Consider a particular disparity (or label) α. A configuration f' is said to be within a single α-expansion move of f if for all pixels $p \in \mathcal{L}$ either $f'_p = f_p$ or $f'_p = \alpha$. Now consider a pair of disparities α, β, $\alpha \neq \beta$. A configuration f' is said to be within a single $\alpha\beta$-swap move of f if for all pixels $p \in \mathcal{L}$, $f_p \notin \{\alpha, \beta\}$ implies $f'_p = f_p$.

The crucial fact about these moves is that for a given configuration f it is possible to efficiently find a strong local minumum of the energy; more precisely, the lowest energy configuration within a single α-expansion or $\alpha\beta$-swap move of f, respectively. These *local improvement operations* rely on graph cuts. The expansion algorithm consists entirely of a sequence of α-expansion local improvement operations for different disparities α, until no α-expansion can reduce the energy. Similarly, the swap algorithm consists entirely of a sequence of $\alpha\beta$-swap local improvement operations for pairs of disparities α, β, until no $\alpha\beta$-swap can reduce the energy.

This formulation, unfortunately, does not handle occlusions properly. First, it can easily happen that two pixels in the left image are mapped

[1] In fact, they consider a more general energy but this is the simplest case that works very well in practice.

into the same pixel in the right image. Furthemore, it assumes that each pixel in the left image is mapped into some pixel in the right image while in reality some pixel in the left image can be occluded and do not correspond to any pixel in the right image.

3 Overview of new algorithms

3.1 Problem representation

Let $\mathcal{A}$ be the set of (unordered) pairs of pixels that may potentially correspond. For stereo with aligned cameras, for example, we have

$$\mathcal{A} = \{ \langle p, q \rangle \mid p_y = q_y \text{ and } 0 \leq q_x - p_x < k \}.$$

(Here we assume that disparities lie in some limited range, so each pixel in $\mathcal{L}$ can potentially correspond to one of k possible pixels in $\mathcal{R}$, and vice versa.) The situation for motion is similar, except that the set of possible disparities is 2-dimensional.

The goal is to find a subset of $\mathcal{A}$ containing only pairs of pixels which correspond to each other. Equivalently, we want to give each assignment $a \in \mathcal{A}$ a value f_a which is 1 if the pixels p and q correspond, and otherwise 0.

Let us define *unique* configurations f. We will call the assignments in $\mathcal{A}$ that have the value 1 *active.* Let $A(f)$ be the set of active assignments according to the configuration f. Let $N_p(f)$ be the set of active assignments in f that involve the pixel p, i.e. $N_p(f) = \{\langle p, q \rangle \in A(f)\}$. We will call a configuration f *unique* if each pixel is involved in at most one active assignment, i.e.

$$\forall p \in \mathcal{P} \quad |N_p(f)| \leq 1.$$

Note that those pixels for which $|N_p(f)| = 0$ are precisely the occluded pixels.

It is possible to extend the notion of α-expansions to our representation. For an assignment $a = \langle p, q \rangle$ let $d(a)$ be its disparity: $d(a) = (q_x - p_x, q_y - p_y)$, and let $\mathcal{A}^\alpha$ be the set of all assignments in $\mathcal{A}$ having disparity α. A configuration f' is said to be within a single α-expansion move of f if $A(f')$ is a subset of $A(f) \cup \mathcal{A}^\alpha$. In other words, some currently active assignments may be deleted, and some assignments having disparity α may be added.

It is also possible to extend the notion of an $\alpha\beta{-}swap$. A configuration f' is said to be within a single $\alpha\beta$-swap move of f if $A(f') \cup \mathcal{A}^{\alpha\beta} = A(f) \cup \mathcal{A}^{\alpha\beta}$, where $\mathcal{A}^{\alpha\beta}$ is the set of all assignments in $\mathcal{A}$ having disparity α or β. In other words, the only changes in f can be adding or deleting assignments having disparities α or β.

3.2 Energy function

Now we define the energy for a configuration f. To correctly handle unique configurations we assume that for non-unique configurations the energy is infinity and for unique configurations the energy is of the form

$$E(f) \quad = \quad E_{data}(f) \ + \ E_{occ}(f) \ + \ E_{smooth}(f). \qquad (2)$$

The three terms here include

- a data term E_{data}, which results from the differences in intensity between corresponding pixels;

- an occlusion term E_{occ}, which imposes a penalty for making a pixel occluded; and

- a smoothness term E_{smooth}, which makes neighboring pixels in the same image tend to have similar disparities.

The data term will be $E_{data}(f) = \sum_{a \in A(f)} D(a)$; typically for an assignment $a = \langle p, q \rangle$, $D(a) = (I(p) - I(q))^2$, where I gives the intensity of a pixel. The occlusion term imposes a penalty C_p if the pixel p is occluded; we will write this as

$$E_{occ}(f) = \sum_{p \in \mathcal{P}} C_p \cdot T(|N_p(f)| = 0).$$

The most nontrivial part here is the choice of smoothness term. It is possible to write several expressions for the smoothness term. The smoothness term involves a notion of neighborhood; we assume that there is a neighborhood system on assignments

$$\mathcal{N} \subset \{ \{a1, a2\} \mid a1, a2 \in \mathcal{A}) \}.$$

One obvious choice is

$$E_{smooth}(f) \;=\; \sum_{\{a1,a2\} \in \mathcal{N}, a1,a2 \in A(f)} V_{a1,a2}, \tag{3}$$

where the neighborhood system $\mathcal{N}$ consists only of pairs $\{a1, a2\}$ such that assignments $a1$ and $a2$ have *different* disparities. $\mathcal{N}$ can include, for example, pairs of assignments $\{\langle p, q \rangle, \langle p', q' \rangle\}$ for which either p and p' are neighbors or q and q' are neighbors, and $d(\langle p, q \rangle) \neq d(\langle p', q' \rangle)$. Thus, we impose a penalty if two close assignments having different disparities are both present in the configuration. Unfortunately, we show in the appendix that not only is minimizing this energy is NP-hard, but also finding a minimum of this function among all configurations within a single α-expansion of the initial configuration is NP-hard as well. As we show in section 6, however, it is possible to efficienly minimize this function over the space of all configurations within a single $\alpha\beta$-swap of the initial configuration.

Our most promising results, however, are obtained using a different smoothness term, which makes it possible to use graph cuts to efficiently find a minimum of the energy among all configurations within a single α-expansion of the initial configuration. The smoothness term is

$$E_{smooth}(f) \;=\; \sum_{\{a1,a2\}\in\mathcal{N}} V_{a1,a2}\cdot T(f(a1)\neq f(a2)). \tag{4}$$

The neighboorhood system here consists only of pairs $\{a1, a2\}$ such that assignments $a1$ and $a2$ have the *same* disparities. It can include, for example, pairs of assignments $\{\langle p,q\rangle, \langle p',q'\rangle\}$ for which p and p' are neighbors, and $d(\langle p,q\rangle) = d(\langle p',q'\rangle)$. Thus, we impose a penalty if one assignment is present in the configuration, and another close assignment, having the same disparity, is not. Although this energy is different from the previous one it enforces the same constraint: if disparities of adjacent pixels are the same then the smoothness penalty is zero, otherwise it has some positive value.

The intuition why this energy allows using graph cuts is the following. It has a similar form to the Potts energy of equation 1. However, it is the Potts energy on *assignments* rather than pixels; as a consequence, none of the previous algorithms based on graph cuts can be applied.

4 Related work

Most work on motion and stereo does not explicitly consider occlusions. For example, correlation based approaches and energy minimization methods based on regularization [14] or Markov Random Fields [10] are typically formulated as labeling problems, where each pixel in one image must be assigned a disparity. This privileges one image over the other, and does not permit occlusions to be naturally incorporated. One common solution with correlation is called cross-checking [5]. This computes disparity twice, both left-to-right and right-to-left, and marks as occlusions those pixels in one image mapping to pixels in the other image which do not map back to them. This method is common and easy to implement, and we will do an experimental comparison against it in section 7.

Similarly, it is possible to incorporate occlusions into energy minimization methods by adding a label that represents being occluded. There are several difficulties, however. It is hard to design a natural energy function that incorporates this new label, and to impose the uniqueness constraint. In addition, these labeling problems still handle the input images asymmetrically.

However, there are a number of papers that elegantly handle occlusions in stereo using energy minimization [2, 4, 9]. These papers focus on computational modeling to understanding the psychophysics of stereopsis; in contrast, we are concerned with accurately computing disparity and occlusion for stereo and motion.

There is one major limitation of the algorithms proposed by [2, 4, 9] which our work overcomes. These algorithms makes extensive use of the ordering constraint, which states that if an object is to the left of another in one stereo

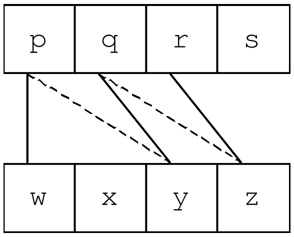


Figure 1: An example of two images with 4 pixels each. Here $\mathcal{L} = \{\texttt{p},\texttt{q},\texttt{r},\texttt{s}\}$ and $\mathcal{R} = \{\texttt{w},\texttt{x},\texttt{y},\texttt{z}\}$. Solid lines indicate the current active assignments, and dashed lines indicated the assignments being considered.

image, it is also to the left in the other image. The advantage of the ordering constraint is efficiency, as it permits the use of dynamic programming. However, the ordering constraint has several limitations. First, depending on the scene geometry, it is not always true. Second, the ordering constraint is specific to stereo, and cannot be used for motion. Third, algorithms that use the ordering constraint essentially solve the stereo problem independently for each scanline. While each scanline can be solved optimally, it is unclear how to impose some kind of inter-scanline consistency. Our method, in contrast, minimizes a natural 2-dimensional energy function, which can be applied to motion as well as to stereo.

Our algorithm is based on graph cuts, which can be used to efficiently minimize a wide range of energy functions. Originally, [11] proved that if there are only two labels the global minimum of the energy can be efficiently computed by a single graph cut. Recent work [6, 13, 16] has shown how to use graph cuts to handle more than two labels. The resulting algorithms have been applied to several problems in early vision, including image restoration and visual correspondence. While graph cuts are a powerful optimization method, the methods of [6, 13, 16] do not handle occlusions gracefully. In addition to all the difficulties just mentioned concerning occlusions and energy minimization, graph cut methods are only applicable to a limited set of energy functions. In particular, previous algorithms cannot be used to minimize the energy E that we define in equation 2.

The most closely related work consists of the recent algorithms based on graph cuts of [12] and [7]. These methods also cannot minimize our energy E. [12] uses graph cuts to explicitly handle occlusions. They handle the input images symetrically and enforce uniqueness. Their graph cut construction actually computes the global minimum in a single graph cut. The limitation of their work lies in the smoothness term, which is the L_1 distance. This smoothness term is not robust, and therefore does not produce good discontinuities. They prove that their construction is only applicable to convex (i.e., non-robust) smoothness terms. In addition, we will prove that minimizing our E is NP-hard, so their construction clearly cannot be applied to our problem.

5 Our expansion move algorithm

We now show how to efficiently minimize E with the smoothness term (4)

$$E_{smooth}(f) \;=\; \sum_{\{a1,a2\}\in\mathcal{N}} V_{a1,a2} \cdot T(f(a1) \neq f(a2)).$$

among all unique configurations using graph cuts. The output of our method will be a local minimum in a strong sense. In particular, consider an input configuration f and a disparity α. Another configuration f' is defined to be

```
1. Start with an arbitrary unique configuration $f$
2. Set success := 0
3. For each disparity $\alpha$
    3.1. Find $\hat{f} = \arg\min E(f')$ among unique $f'$ within single
         $\alpha$-expansion of $f$
    3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return $f$
```

Figure 2: The steps of the expansion algorithm

within a single α-*expansion* of f if some assignments in f become inactive, and some assignments with disparity α become active (a formal definition is given at the start of section 5.2.1).

Our algorithm is very straightforward (figure 2); we simply select (in a fixed order or at random) a disparity α, and we find the unique configuration within a single α-expansion move (our local improvement step). If this decreases the energy, then we go there; if there is no α that decreases the energy, we are done. The critical step in our method is to efficiently compute the α-expansion with the smallest energy. In this section, we show how to use graph cuts to solve this problem.

5.1 Graph cuts

Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ be a weighted graph with two distinguished terminal vertices $\{s, t\}$ called the source and sink. A *cut* $\mathcal{C} = \mathcal{V}^s, \mathcal{V}^t$ is a partition of the

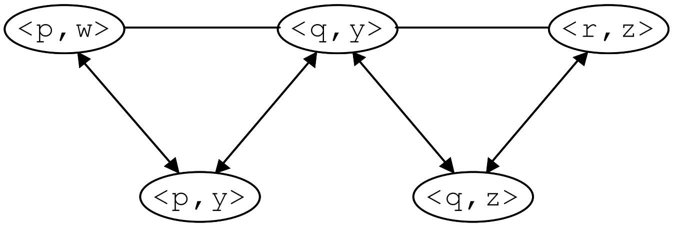


Figure 3: The graph corresponding to figure 1. There are links between all vertices and the terminals, which are not shown. Edges without arrows are bidirectional edges with the same weight in each direction; edges with arrows have different weights in each direction.

vertices into two sets such that $s \in \mathcal{V}^s$ and $t \in \mathcal{V}^t$.[2] The cost of the cut, denoted $|\mathcal{C}|$, equals the sum of the weights of the edges between a vertex in $\mathcal{V}^s$ and a vertex in $\mathcal{V}^t$.

The minimum cut problem is to find the cut with the smallest cost. This problem can be solved very efficiently by computing the maximum flow between the terminals, according to a theorem due to Ford and Fulkerson [8]. There are a large number of fast algorithms for this problem (see [1], for example). The worst case complexity is low-order polynomial; however, in practice the running time is nearly linear for graphs with many short paths between the source and the sink, such as the one we will construct.

5.2 Computing a local minimum

We first construct the graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, and give the correspondence between cuts on $\mathcal{G}$ and configurations. Then we show that the minimum cut on

[2] A cut can also be equivalently defined as the set of edges between the two sets.

$\mathcal{G}$ yields the configuration that minimizes E among unique configurations within one α-expansion.

5.2.1 Graph structure

In an α-expansion, active assignments may become inactive, and inactive assignments whose disparity is α may become active. Suppose that we start off with a unique configuration f^0. The active assignments for a new configuration within one α-expansion will be a subset of $\tilde{A} = \mathcal{A}^0 \cup \mathcal{A}^\alpha$, where $\mathcal{A}^0 = \{\, a \in A(f^0) \mid d(a) \neq \alpha \,\}$ and $\mathcal{A}^\alpha = \{\, a \in \mathcal{A} \mid d(a) = \alpha \,\}$. We will define the configuration $\tilde{f}$ by $A(\tilde{f}) = \tilde{A}$. Note that in general $\tilde{f}$ is not unique.

The directed graph $\mathcal{G}$ that we will construct has vertices that correspond to assignments; this is in contrast to the graphs built by [6, 7, 13, 16]. The terminals will be called s and t, and for every assignment in $\tilde{A}$ there will be a vertex.

The edges in $\mathcal{G}$ are as follows. For every vertex $a \in \tilde{A}$ there will be edges (s, a) and (a, t). In addition, if $\{a1, a2\} \in \mathcal{N}$ there will be edges $(a1, a2)$ and $(a2, a1)$. Note that in this case, either $a1$ and $a2$ are both in $\mathcal{A}^0$ or they are both in $\mathcal{A}^\alpha$. Finally, consider a pair of vertices $a1, a2$ that enter a common pixel p (i.e., where $a1 = \langle p, q\rangle$ and $a2 = \langle p, r\rangle$). Note that in this case either $a1 \in \mathcal{A}^0, a2 \in \mathcal{A}^\alpha$ or vice-versa. There will be edges between every such pair of assignments.

Now consider a cut $\mathcal{C} = \mathcal{V}^s, \mathcal{V}^t$ on $\mathcal{G}$. The configuration $f^{\mathcal{C}}$ that corresponds to this cut is defined by

$$\forall a \in \mathcal{A}^0 \qquad f_a^{\mathcal{C}} = \begin{cases} 1 & \text{if } a \in \mathcal{V}^s \\ 0 & \text{if } a \in \mathcal{V}^t \end{cases}$$

$$\begin{aligned} \forall a \in \mathcal{A}^{\alpha} \qquad f_a^{\mathcal{C}} &= \begin{cases} 1 & \text{if } a \in \mathcal{V}^t \\ 0 & \text{if } a \in \mathcal{V}^s. \end{cases} \\ \forall a \notin \tilde{A} \qquad f_a^{\mathcal{C}} &= 0 \end{aligned} \tag{5}$$

The following lemma is an obvious consequence of this construction.

Lemma 5.1 *$\mathcal{C}$ is a cut on $\mathcal{G}$ if and only if the configuration $f^{\mathcal{C}}$ lies within a single α-expansion of the input configuration f^0.*

We now give the weights of the edges in $\mathcal{G}$. First, we define the occlusion cost

$$D_{occ}(\langle p, q \rangle) = D_{occ}(p) + D_{occ}(q),$$

where $D_{occ}(p) = C_p$ if $\tilde{A}$ has only one edge entering p, and 0 otherwise. We define the smoothness cost by

$$D_{smooth}(a1) = \sum_{\substack{\{a1,a2\} \in \mathcal{N} \\ a2 \notin \tilde{A}}} V_{a1,a2}.$$

Then the weights are as follows.

edge	**weight**	**for**
(s, a)	$D_{occ}(a)$	$a \in \mathcal{A}^0$
(a, t)	$D_{occ}(a)$	$a \in \mathcal{A}^{\alpha}$
(a, t)	$D(a) + D_{smooth}(a)$	$a \in \mathcal{A}^0$
(s, a)	$D(a)$	$a \in \mathcal{A}^{\alpha}$
$(a1,a2)$ $(a2,a1)$	$V_{a1,a2}$	$\{a1,a2\} \in \mathcal{N}$, $a1,a2 \in \tilde{A}$
$(a1, a2)$	∞	$p \in \mathcal{P}, a1 \in \mathcal{A}^0, a2 \in \mathcal{A}^{\alpha}$ $a1,a2 \in N_p(\tilde{f})$
$(a2, a1)$	C_p	$p \in \mathcal{P}, a1 \in \mathcal{A}^0, a2 \in \mathcal{A}^{\alpha}$ $a1,a2 \in N_p(\tilde{f})$

We will refer to the links with weight $D_{occ}(a)$ (i.e., the top two rows of the above table) as *t-links*. We will refer to the links with cost C_p as *c-links*.

A small example is shown in figure 1. The current set of assignments is shown with solid lines; dashed lines represent the new assignments we are considering (i.e., $\alpha = 2$). In the current configuration, the pixels `s` and `x` are occluded, and the proposed expansion move will not change their status.

The corresponding graph is shown in figure 3. The 3 nodes in the top row form $\mathcal{A}^0$ and the two nodes in the bottom row form $\mathcal{A}^\alpha$. Note, for example, that the edge from $\langle p, w\rangle$ to $\langle p, y\rangle$ has weight ∞, since these two assignments cannot both be active.

5.2.2 Optimality

We now show that if $\mathcal{C}$ is the minimum cut on our graph $\mathcal{G}$, then $f^{\mathcal{C}}$ is the configuration that minimizes the energy E over unique configurations.

Lemma 5.2 *The cost of the cut $\mathcal{C}$ is finite if and only if the corresponding configuration $f^{\mathcal{C}}$ is unique.*

PROOF: If $f^{\mathcal{C}}$ is not unique there is some pixel $p \in \mathcal{P}$ such that a pair of assignments $a1, a2 \in N_p(f^{\mathcal{C}})$ are both in $A(f^{\mathcal{C}})$. Without loss of generality let $a1 \in \mathcal{A}^0$ and $a2 \in \mathcal{A}^\alpha$. Then we have $a1 \in \mathcal{V}^s$ and $a2 \in \mathcal{V}^t$, so the edge $(a1, a2)$, which has weight ∞, must be cut. Similarly, if the weight of $\mathcal{C}$ is infinite, one of these edges is cut, so some pixel p is not unique. ■

Lemma 5.3 *Let $f^{\mathcal{C}}$ be a unique configuration, with corresponding cut $\mathcal{C}$. Then the cost of the t-links plus the c-links in $\mathcal{C}$ equals $E_{occ}(f^{\mathcal{C}})$ plus a constant.*

PROOF: The cost of the t-links is

$$\sum_{a\in\mathcal{A}^0} D_{occ}(a)\cdot T(a\in\mathcal{V}^t)+\sum_{a\in\mathcal{A}^\alpha} D_{occ}(a)\cdot T(a\in\mathcal{V}^s). \tag{6}$$

The cost of the c-links is

$$\sum_{\substack{p\in\mathcal{P},a1\in\mathcal{A}^0,a2\in\mathcal{A}^\alpha\\ a1,a2\in N_p(\tilde{f})}} C_p\cdot T(a1\in\mathcal{V}^t\wedge a2\in\mathcal{V}^s). \tag{7}$$

We also have

$$E_{occ}(f^{\mathcal{C}})=\sum_{p\in\mathcal{P}} C_p\cdot T(|N_p(f^{\mathcal{C}})|=0)$$

which is a constant plus

$$\sum_{\substack{p\in\mathcal{P}\\ |N_p(\tilde{f})|=1}} C_p\cdot T(|N_p(f^{\mathcal{C}})|=0)+\sum_{\substack{p\in\mathcal{P}\\ |N_p(\tilde{f})|=2}} C_p\cdot T(|N_p(f^{\mathcal{C}})|=0).$$

We can write this as a constant plus the three terms

$$\begin{aligned}&\sum_{\substack{p\in\mathcal{P}\\ N_p(\tilde{f})=\{a\}\subset\mathcal{A}^0}} D_{occ}(p)\cdot T(a\notin A(f^{\mathcal{C}}))\\ &+\sum_{\substack{p\in\mathcal{P}\\ N_p(\tilde{f})=\{a\}\subset\mathcal{A}^\alpha}} D_{occ}(p)\cdot T(a\notin A(f^{\mathcal{C}}))\\ &+\sum_{\substack{p\in\mathcal{P},a1\in\mathcal{A}^0,a2\in\mathcal{A}^\alpha\\ a1,a2\in N_p(\tilde{f})}} C_p\cdot T(a1,a2\notin A(f^{\mathcal{C}})).\end{aligned}$$

This equals the sum of equations 6 and 7. ■

Theorem 5.4 *Let $\mathcal{C}$ be the minimum cut on $\mathcal{G}$. Then $f^{\mathcal{C}}$ is the unique configuration within one α-expansion of f^0 that minimizes the energy E.*

PROOF: Lemma 5.1 shows that $f^{\mathcal{C}}$ lies within one α-expansion of f^0. Lemma 5.2 shows that the minimum cut is unique, since there are obviously cuts on $\mathcal{G}$

with finite costs; therefore, the links with infinite cost are not included in $\mathcal{C}$. Due to lemma 5.3, all that remains is to show that the cost of $\mathcal{C}$, ignoring t-links and c-links, is $E_{data}(f^{\mathcal{C}}) + E_{smooth}(f^{\mathcal{C}})$, which is

$$\sum_{a \in A(f^{\mathcal{C}})} D(a) + \sum_{\{a1,a2\} \in \mathcal{N}} V_{a1,a2} \cdot T(f^{\mathcal{C}}_{a1} \neq f^{\mathcal{C}}_{a2}).$$

The second sum can be rewritten as

$$\sum_{\substack{\{a1,a2\} \in \mathcal{N} \\ a1,a2 \in \tilde{A}}} V_{a1,a2} \cdot T(f^{\mathcal{C}}_{a1} \neq f^{\mathcal{C}}_{a2}) + \sum_{\substack{\{a1,a2\} \in \mathcal{N} \\ a1 \in \tilde{A}, a2 \notin \tilde{A}}} V_{a1,a2} \cdot T(f^{\mathcal{C}}_{a1} \neq f^{\mathcal{C}}_{a2}).$$

Ignoring t-links and c-links, the cost of $\mathcal{C}$ is

$$\sum_{a \in \mathcal{A}^0} (D(a) + D_{smooth}(a)) \cdot T(a \in \mathcal{V}^s)$$

$$+ \sum_{a \in \mathcal{A}^\alpha} D(a) \cdot T(a \in \mathcal{V}^t)$$

$$+ \sum_{\substack{\{a1,a2\} \in \mathcal{N} \\ a1,a2 \in \tilde{A}}} V_{a1,a2} \cdot T((a1 \in \mathcal{V}^s, a2 \in \mathcal{V}^t) \vee (a1 \in \mathcal{V}^t, a2 \in \mathcal{V}^s)).$$

The first two terms are

$$\sum_{a \in \mathcal{A}^0 \cap A(f^{\mathcal{C}})} (D(a) + D_{smooth}(a)) + \sum_{a \in \mathcal{A}^\alpha \cap A(f^{\mathcal{C}})} D(a),$$

while the third is simply

$$\sum_{\substack{\{a1,a2\} \in \mathcal{N} \\ a1,a2 \in \tilde{A}}} V_{a1,a2} \cdot T(f^{\mathcal{C}}_{a1} \neq f^{\mathcal{C}}_{a2}).$$

The terms involving $D(a)$ sum to $\sum_{a \in A(f^{\mathcal{C}})} D(a)$, so all we need is to show

$$\sum_{\substack{\{a1,a2\} \in \mathcal{N} \\ a1 \in \tilde{A}, a2 \notin \tilde{A}}} V_{a1,a2} \cdot T(f^{\mathcal{C}}_{a1} \neq f^{\mathcal{C}}_{a2}) = \sum_{a \in \mathcal{A}^0 \cap A(f^{\mathcal{C}})} D_{smooth}(a).$$

In the first expression, $a1 \in \mathcal{A}^0$, since $a1 \in \mathcal{A}^\alpha$ and $\{a1, a2\} \in \mathcal{N}$ imply $a2 \in \mathcal{A}^\alpha \subset \tilde{A}$. The proof now follows from the definition of D_{smooth}. ■

```
1. Start with an arbitrary configuration $f$
2. Set success := 0
3. For each pair of disparities $\alpha$, $\beta$ ($\alpha \neq \beta$)
    3.1. Find $\hat{f} = \arg\min E(f')$ among $f'$ within one $\alpha\beta$-swap of $f$
    3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return $f$
```

Figure 4: The steps of the swap algorithm

6 Our swap-move algorithm

In this section we present another algorithm minimizing the energy with the smoothness term (3)

$$E_{smooth}(f) \;=\; \sum_{\{a1,a2\}\in\mathcal{N},a1,a2\in A(f)} V_{a1,a2}$$

Note that although it is possible to enforce the uniqueness constraint by setting $V_{a1,a2} = \infty$ for assignments that enter the common pixel p, i.e. $a1 = \langle p, q' \rangle$ and $a2 = \langle p, q'' \rangle$, it is not necessary for the construction to work, in the constrast to the expansion algorithm. Thus, we will not assume that configurations are unique.

The general structure of the swap algorithm is similar to the one of the expansion algorithm.

As for the expansion algorithm, the critical step is 3.1 - minimizing the energy over the space of configurations within one $\alpha\beta$-swap of f. In the next section we give the details of the graph consruction for $\alpha\beta$-swap.

6.1 Graph structure

Suppose that we start off with a configuration f^0. Let $\mathcal{A}^0$ be the set of active assignments of the configuration f^0 that have disparities different from α and β. In an $\alpha\beta$-swap, assignments having disparities α or β may change their status. Let $\mathcal{A}^\alpha = \{\, a \in \mathcal{A} \mid d(a) = \alpha \,\}$, $\mathcal{A}^\beta = \{\, a \in \mathcal{A} \mid d(a) = \beta \,\}$ and $\mathcal{A}^{\alpha\beta} = \mathcal{A}^\alpha \cup \mathcal{A}^\beta$. The active assignments for a new configuration within one $\alpha\beta$-swap will be a subset of $\tilde{A} = \mathcal{A}^0 \cup \mathcal{A}^{\alpha\beta}$ containing $\mathcal{A}^0$.

The directed graph $\mathcal{G}$ that we will construct has vertices that correspond to all assignments in $\mathcal{A}^{\alpha\beta}$, as well as the terminals s and t.

The edges in $\mathcal{G}$ are as follows. For every vertex $a \in \mathcal{A}^{\alpha\beta}$ there will be edges (s, a) and (a, t). In addition, if $a1 \in \mathcal{A}^\alpha$, $a2 \in \mathcal{A}^\beta$ and $\{a1, a2\} \in \mathcal{N}$ there will be an edge $(a1, a2)$. Finally, consider a pair of vertices $a1, a2$ that enter a common pixel p (i.e., where $a1 = \langle p, q\rangle$ and $a2 = \langle p, r\rangle$) and no assignment in $\mathcal{A}^0$ enters p. Note that in this case one of the assignments is in $\mathcal{A}^\alpha$, and the other one is in $\mathcal{A}^\beta$; let $a1 \in \mathcal{A}^0, a2 \in \mathcal{A}^\alpha$. There will be an edge $(a2, a1)$ for every such pair of assignments.

Now consider a cut $\mathcal{C} = \mathcal{V}^s, \mathcal{V}^t$ on $\mathcal{G}$. The configuration $f^{\mathcal{C}}$ that corresponds to this cut is defined by

$$\begin{aligned}
\forall a \in \mathcal{A}^\alpha \qquad f^{\mathcal{C}}_a &= \begin{cases} 1 & \text{if } a \in \mathcal{V}^s \\ 0 & \text{if } a \in \mathcal{V}^t \end{cases} \\
\forall a \in \mathcal{A}^\beta \qquad f^{\mathcal{C}}_a &= \begin{cases} 1 & \text{if } a \in \mathcal{V}^t \\ 0 & \text{if } a \in \mathcal{V}^s. \end{cases} \\
\forall a \notin \mathcal{A}^{\alpha\beta} \qquad f^{\mathcal{C}}_a &= {f^0}_a
\end{aligned} \tag{8}$$

The following lemma is an obvious consequence of this construction.

Lemma 6.1 *$\mathcal{C}$ is a cut on $\mathcal{G}$ if and only if the configuration $f^{\mathcal{C}}$ lies within a single $\alpha\beta$-swap of the input configuration f^0.*

We now give the weights of the edges in $\mathcal{G}$. First, we define the occlusion cost

$$D_{occ}(\langle p, q\rangle) = D_{occ}(p) + D_{occ}(q),$$

where $D_{occ}(p) = C_p$ if $\tilde{A}$ has only one edge entering p, and 0 otherwise. We define the smoothness cost by

$$D_{smooth}(a1) = \sum_{\substack{\{a1,a2\}\in\mathcal{N} \\ a2\in\mathcal{A}^0}} V_{a1,a2}$$

Then the weights are as follows.

edge	**weight**	**for**
(s, a)	$D_{occ}(a)$	$a \in \mathcal{A}^\alpha$
(a, t)	$D_{occ}(a)$	$a \in \mathcal{A}^\beta$
(a, t)	$D(a) + D_{smooth}(a)$	$a \in \mathcal{A}^\alpha$
(s, a)	$D(a) + D_{smooth}(a)$	$a \in \mathcal{A}^\beta$
$(a1, a2)$	$V_{a1,a2}$	$\{a1,a2\}\in\mathcal{N}$, $a1\in\mathcal{A}^\alpha, a2\in\mathcal{A}^\beta$
$(a2, a1)$	C_p	$p\in\mathcal{P}, a1\in\mathcal{A}^\alpha, a2\in\mathcal{A}^\beta$ $N_p(\tilde{f})=\{a1,a2\}$

The configuration $\tilde{f}$ in the last row is the one corresponding to the set $\tilde{A}$; condition $N_p(\tilde{f}) = \{a1, a2\}$ means that assignments $a1$ and $a2$ are the only assignments in $\tilde{A}$ entering the pixel p.

We will refer to the links with weight $D_{occ}(a)$ (i.e., the top two rows of the above table) as *t-links*. We will refer to the links with cost C_p as *c-links*.

6.2 Optimality

We now show that if $\mathcal{C}$ is the minimum cut on our graph $\mathcal{G}$, then $f^{\mathcal{C}}$ is the configuration that minimizes the energy E over configurations within one $\alpha\beta$-swap.

Lemma 6.2 *Let $f^{\mathcal{C}}$ be a configuration that corresponds to a cut $\mathcal{C}$. Then the cost of the t-links plus the c-links in $\mathcal{C}$ equals $E_{occ}(f^{\mathcal{C}})$ plus a constant.*

Proof: The cost of the t-links is

$$\sum_{a\in\mathcal{A}^{\alpha}} D_{occ}(a)\cdot T(a\in\mathcal{V}^t)+\sum_{a\in\mathcal{A}^{\beta}} D_{occ}(a)\cdot T(a\in\mathcal{V}^s). \tag{9}$$

The cost of the c-links is

$$\sum_{\substack{p\in\mathcal{P},a1\in\mathcal{A}^{\alpha},a2\in\mathcal{A}^{\beta}\\ N_p(\tilde{f})=\{a1,a2\}}} C_p\cdot T(a1\in\mathcal{V}^t\wedge a2\in\mathcal{V}^s). \tag{10}$$

We also have

$$E_{occ}(f^{\mathcal{C}})=\sum_{p\in\mathcal{P}^{occ}} C_p\cdot T(|N_p(f^{\mathcal{C}})|=0)$$

where $\mathcal{P}^{occ}$ is the set of pixels p in $\mathcal{P}$ such that no active assignments in $\mathcal{A}^0$ enter p; pixels which are not in $\mathcal{P}^{occ}$ will not be occluded in $f^{\mathcal{C}}$. The last expression is a constant plus

$$\sum_{\substack{p\in\mathcal{P}^{occ}\\ |N_p(\tilde{f})|=1}} C_p\cdot T(|N_p(f^{\mathcal{C}})|=0)+\sum_{\substack{p\in\mathcal{P}^{occ}\\ |N_p(\tilde{f})|=2}} C_p\cdot T(|N_p(f^{\mathcal{C}})|=0).$$

Note that $p\in\mathcal{P}^{occ}$ if and only if $N_p(\tilde{f})\subset\mathcal{A}^{\alpha\beta}$. Thus, we can rewrite the last expression as the three terms

$$\sum_{\substack{p\in\mathcal{P}\\ N_p(\tilde{f})=\{a\}\subset\mathcal{A}^{\alpha}}} D_{occ}(p)\cdot T(a\notin A(f^{\mathcal{C}}))$$

$$+ \sum_{\substack{p\in\mathcal{P} \\ N_p(\tilde{f})=\{a\}\subset\mathcal{A}^\beta}} D_{occ}(p)\cdot T(a\notin A(f^{\mathcal{C}}))$$

$$+ \sum_{\substack{p\in\mathcal{P}, a1\in\mathcal{A}^\alpha, a2\in\mathcal{A}^\beta \\ N_p(\tilde{f})=\{a1,a2\}}} C_p\cdot T(a1,a2\notin A(f^{\mathcal{C}})).$$

This equals the sum of equations 9 and 10. ■

Theorem 6.3 *Let $\mathcal{C}$ be the minimum cut on $\mathcal{G}$. Then $f^{\mathcal{C}}$ is the configuration within one $\alpha\beta$-swap of f^0 that minimizes the energy E.*

PROOF: Lemma 6.1 shows that $f^{\mathcal{C}}$ lies within one $\alpha\beta$-expansion of f^0. Due to lemma 6.2, all that remains is to show that the cost of $\mathcal{C}$, ignoring t-links and c-links, is $E_{data}(f^{\mathcal{C}})+E_{smooth}(f^{\mathcal{C}})$, which is

$$\sum_{a\in A(f^{\mathcal{C}})} D(a) + \sum_{\{a1,a2\}\in\mathcal{N}} V_{a1,a2}\cdot T(a1,a2\in A(f^{\mathcal{C}})).$$

The second sum can be rewritten as a constant plus

$$\sum_{\substack{\{a1,a2\}\in\mathcal{N} \\ a1\in\mathcal{A}^{\alpha\beta}, a2\in\mathcal{A}^0}} V_{a1,a2}\cdot T(a1,a2\in A(f^{\mathcal{C}})).$$

$$+ \sum_{\substack{\{a1,a2\}\in\mathcal{N} \\ a1,a2\in\mathcal{A}^{\alpha\beta}}} V_{a1,a2}\cdot T(a1,a2\in A(f^{\mathcal{C}})) \tag{11}$$

Ignoring t-links and c-links, the cost of $\mathcal{C}$ is

$$\sum_{a\in\mathcal{A}^\alpha} (D(a)+D_{smooth}(a))\cdot T(a\in\mathcal{V}^s)$$

$$+ \sum_{a\in\mathcal{A}^\beta} (D(a)+D_{smooth}(a))\cdot T(a\in\mathcal{V}^t)$$

$$+ \sum_{\substack{\{a1,a2\}\in\mathcal{N} \\ a1\in\mathcal{A}^\alpha, a2\in\mathcal{A}^\beta}} V_{a1,a2}\cdot T(a1\in\mathcal{V}^s, a2\in\mathcal{V}^t). \tag{12}$$

The terms involving $D(a)$ sum to a constant plus $\sum_{a\in A(f^C)} D(a)$. The sum of the terms involving D_{smooth} equals the first term in equation 11. The last term in the equation 12 equals the last term in the equation 11.

$\blacksquare$

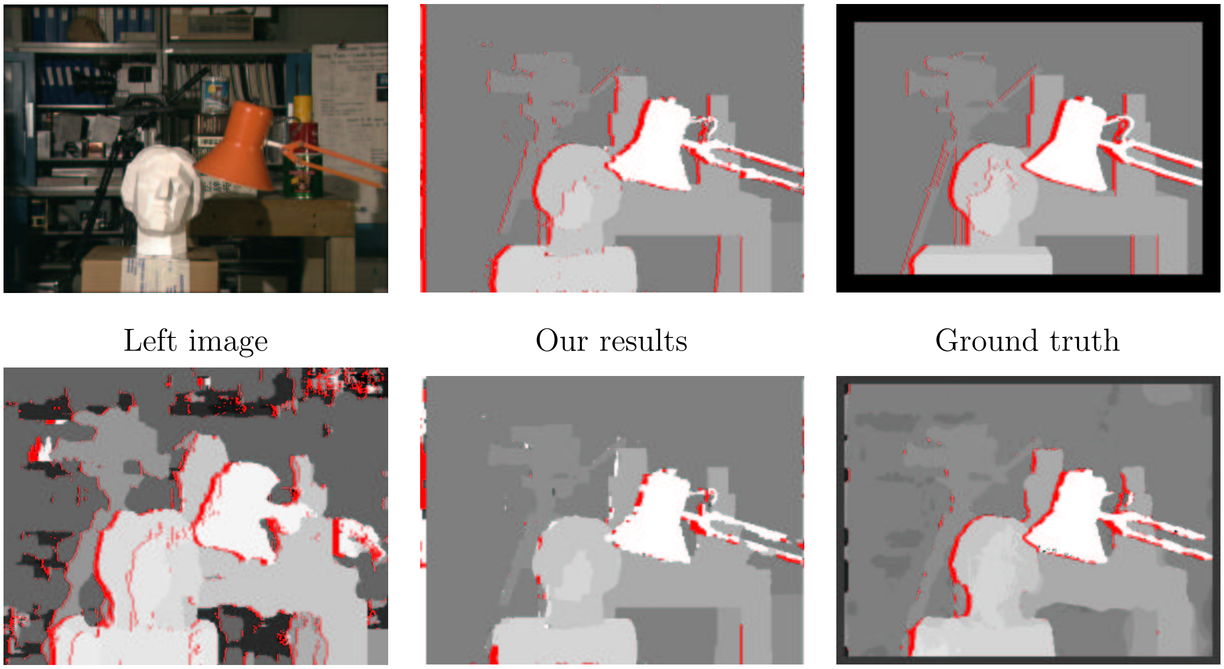
7 Experimental results

Our experimental results involve both stereo and motion. The expansiom ove algorithm gives much higher quality results than the swap move algorithm, so we have focussed on it (at the end of this section we show a result from the swap move algorithm). Our optimization method does not have any parameters except for the exact choice of E. We selected the labels α in random order, and we started with an initial solution in which no assignments are active. For our data term D we made use of the method of Birchfield and Tomasi [3] to handle sampling artifacts. The choice of $V_{a1,a2}$ was designed to make it more likely that a pair of adjacent pixels in one image with similar intensities would end up with similar disparities. If $a1 = \langle p, q\rangle$ and $a2 = \langle r, s\rangle$, then $V_{a1,a2}$ was implemented as an empirically selected decreasing function of $\max(|I(p) - I(r)|, |I(q) - I(s)|)$ as follows:

$$V_{a1,a2} = \begin{cases} \lambda & \text{if } \max(|I(p) - I(r)|, |I(q) - I(s)|) < 8, \\ 3\lambda & \text{otherwise.} \end{cases} \tag{13}$$

The occlusion penalty was chosen to be 2.5λ for all pixels. Thus, the energy depends only on one parameter λ. For different images we picked λ empirically.

We compared the results with the expansion algorithm described in [7]



Left image

Our results

Ground truth

L_1 correlation

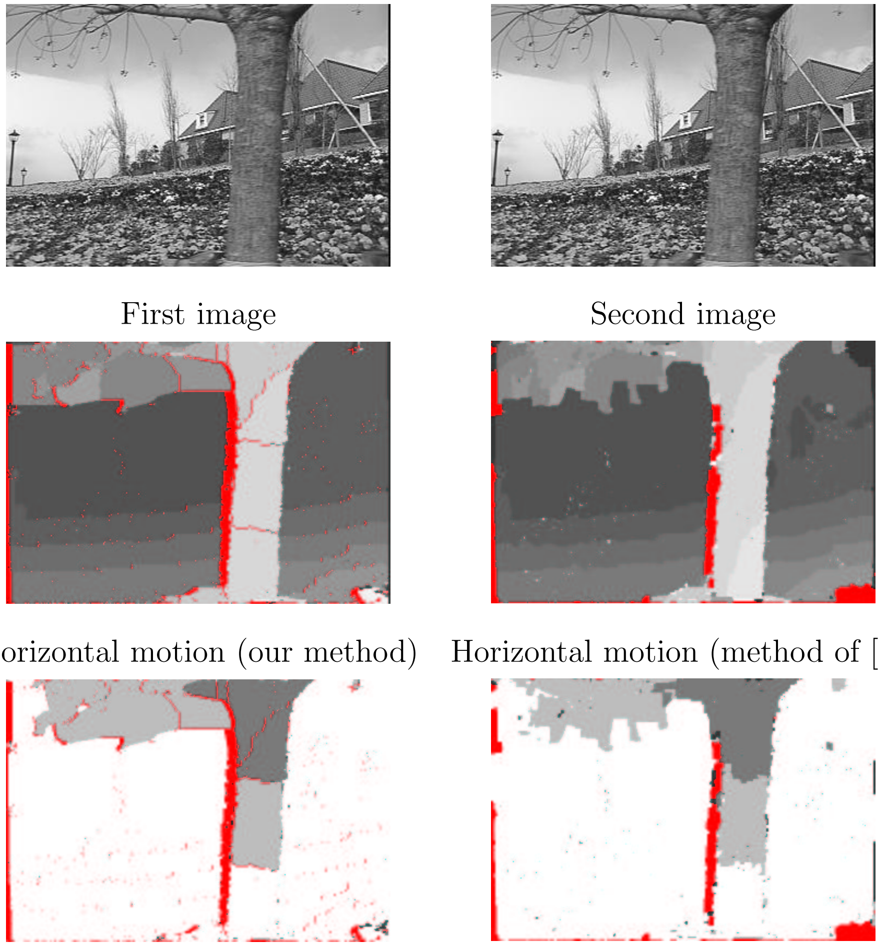
Results from [7]

Results from [19]

Figure 5: Stereo results on Tsukuba dataset. Occluded pixels are shown in red.

Method	Errors	Gross errors	False negatives	False positives
Our results	6.7%	1.9%	42.6%	1.1%
Our results (swap algorithm)	20.7%	13.6%	50.6%	3.4%
Boykov, Veksler & Zabih [7]	6.7%	2.0%	82.8%	0.3%
Zitnick & Kanade [19]	12.0%	2.6%	52.4%	0.8%
Correlation	28.5%	12.8%	87.3%	6.1%

Figure 6: Error statistics on Tsukuba dataset.



First image

Second image

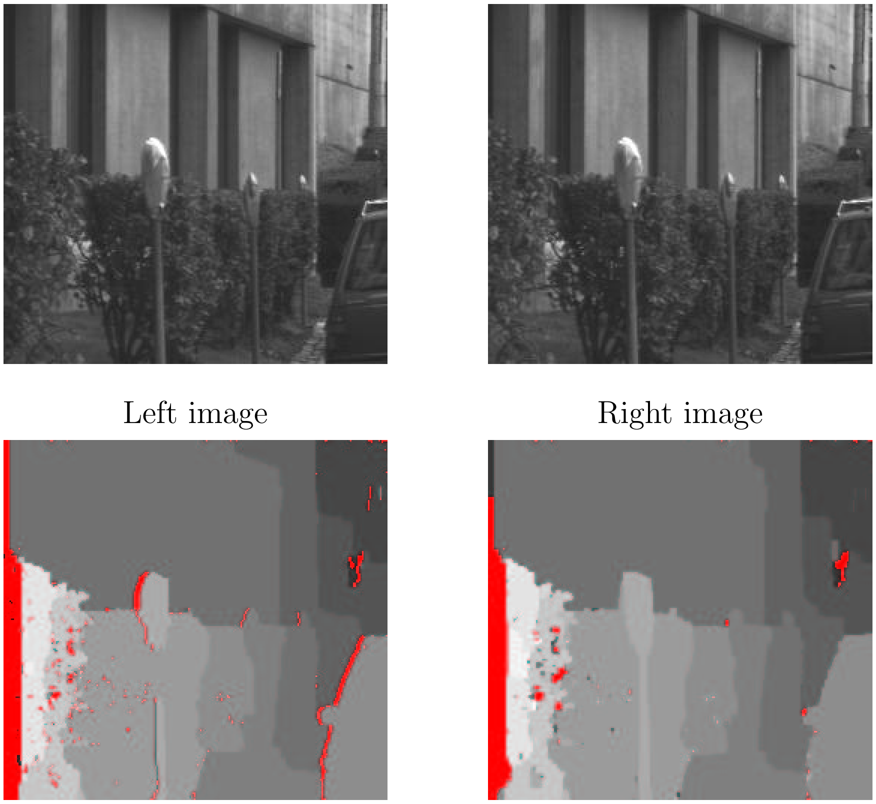
Horizontal motion (our method)

Horizontal motion (method of [7])

Vertical motion (our method)

Vertical motion (method of [7])

Figure 7: Motion results on the flower garden sequence. Occluded pixels are shown in red.



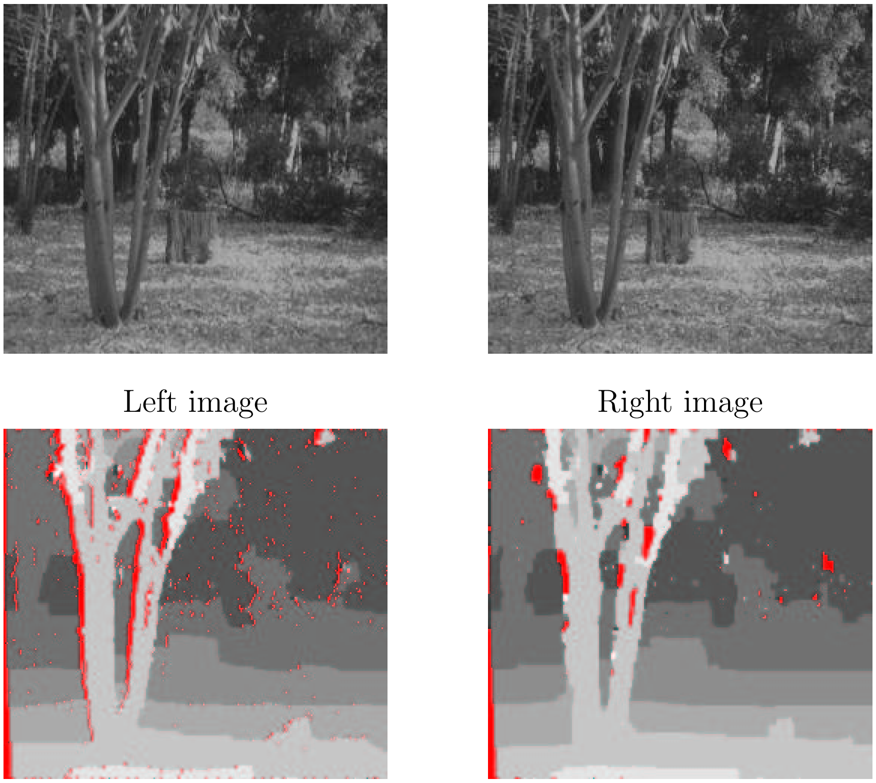
Left image

Right image

Horizontal motion (our method)

Horizontal motion (method of [7])

Figure 8: Stereo results on the meter image. Occluded pixels are shown in red.



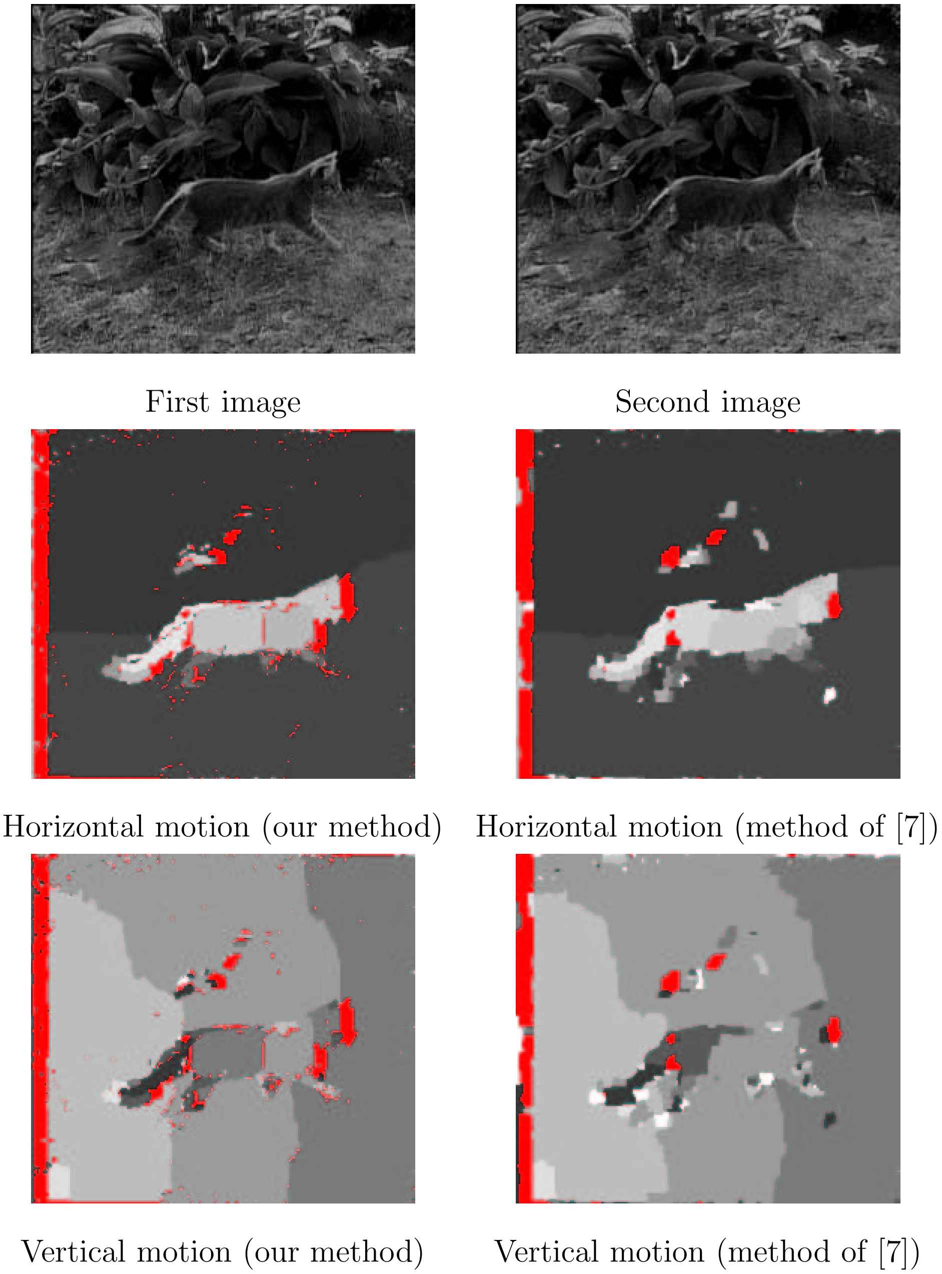
Left image

Right image

Horizontal motion (our method)

Horizontal motion (method of [7])

Figure 9: Stereo results on the SRI tree sequence. Occluded pixels are shown in red.



First image

Second image

Horizontal motion (our method)

Horizontal motion (method of [7])

Vertical motion (our method)

Vertical motion (method of [7])

Figure 10: Motion results on the cat sequence. Occluded pixels are shown in red.

with the additional explicit label 'occluded', since this is the closest related work. For the data with ground truth we obtained some recent results due to Zitnick and Kanade [19]. We also implemented correlation using the L_1 distance. Occlusions were computed using cross-checking, which computes matches left-to-right and right-to-left, and then marks a pixel as occluded if it maps to a pixel that does not map back to it. We used a 13 by 13 window for correlation; we experimented with several other window sizes and other variants of correlation, but they all gave comparable results.

Quantitative comparison of various methods was made on a stereo image pair from the University of Tsukuba with hand-labeled integer disparities. The left input image and the ground truth are shown in figure 5, together with our results and the results of various other methods. The Tsukuba images are 384 by 288; in all the experiments with this image pair we used 16 disparities.

We have computed the error statistics, which are shown in figure 6. We used the ground truth to determine which pixels are occluded. For the first two columns, we ignored the pixels that are occluded in the ground truth. We determined the percentage of the remaining pixels where the algorithm did not compute the correct disparity (the "Errors" column), or a disparity within ±1 of the correct disparity ("Gross errors"). We considered labeling a pixel as occluded to be a gross error. The last two columns show the error rates for occlusions.

We have also experimented with a number of standard sequences. The results from the flower garden (motion) sequence are shown in figure 7, and the CMU meter and SRI tree (stereo) results are shown in figures 8 and 9.

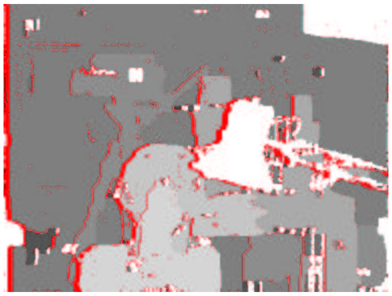
For comparison we have shown the results from the expansion algorithm of [7]. In addition, results are shown in figure 10 for a very challenging sequence involving the non-rigid motion of a kitten in a windy garden.

The running times for our algorithm are on average about 25% slower than the expansion algorithm of [7], but on the order of a minute. For example, on the Tsukuba data set our algorithm takes 83 seconds, while [7] takes 75 seconds. These numbers were obtained using a 500 Megahertz Pentium-III.

We have also experimented with the parameter sensitivity of our method. Since there is only one parameter, namely λ in equation 13, it is easy to experimentally determine the algorithm's sensitivity. The table below shows that our method is relatively insensitive to the exact choice of λ.

λ	1	3	10	30
Error	10.9%	6.7%	9.7%	11.1%
Gross errors	2.4%	1.9%	3.1%	3.6%
False neg.	42.2%	42.6%	48.0%	51.4%
False pos.	1.4%	1.1%	1.0%	0.8%

We have also implemented the swap move algorithm described in section 6. The work of [7] suggested that swap moves and expansion moves give fairly comparable results. However, with our (more complex) energy functions, the expansion move algorithm gives significantly better results. For the sake of completeness, the output of the swap move algorithm on the Tsukuba imagery is given below.



It appears that swap moves are simply not powerful enough to escape local minima for this class of energy functions.

8 Conclusions

We have presented an energy minimization formulation of the correspondence problem with occlusions, and given a fast approximation algorithm based on graph cuts. The experimental results for both stereo and motion appear promising. Our method can easily be generalized to associate a cost with labeling a particular assignment as inactive.

Acknowledgements

We thank Y. Ohta and Y. Nakamura for supplying the ground truth imagery from the University of Tsukuba, L. Zitnick for providing us with the output of his method, and O. Veksler for useful comments.

Appendix A: Finding a local minimum of E with the smoothness term (3) within a single α-expansion is NP-hard

Let's call the problem of finding a local minimum of the energy in equation 2 with the smoothness term term (3) within a single α-expansion an *expansion* problem. In this section we will show that this is an NP-hard problem by reducing the independent set problem, which is known to be NP-hard, to the expansion problem.

Let an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the input to the independent set problem. The subset $\mathcal{U} \subset \mathcal{V}$ is said to be independent if for any two nodes $u, v \in \mathcal{U}$ the edge (u, v) is not in $\mathcal{E}$. The goal is to find an independent subset $\mathcal{U}^* \subset \mathcal{V}$ of maximum cardinality. We construct an instance of the expansion problem as follows. For each node $v \in \mathcal{V}$ we create pixels $l(v) \in \mathcal{L}$ in the left image, $r(v) \in \mathcal{R}$ in the right image and the assignment $a(v) = \langle l(v), r(v) \rangle \in \mathcal{A}$ in such a way that disparities for different assignments $a(u)$ and $a(v)$ are different $(u, v \in \mathcal{V},\ u \neq v)$. Thus, we have $|\mathcal{L}| = |\mathcal{R}| = |\mathcal{A}| = |\mathcal{V}|$.

The neighboring system $\mathcal{N}$ on assignments will be constructed from the connectivity of the graph $\mathcal{G}$: for every edge $(u, v) \in \mathcal{E}$ we add the pair of assigments $\{a(u), a(v)\}$ to $\mathcal{N}$. The corresponding penalty for a discontinuity will be $V_{a(u),a(v)} = C$, where C is a sufficiently large constant $(C > |\mathcal{V}|)$. The data term will be 0 for all assignments, and the occlusion penalty will be $\frac{1}{2}$ for all pixels.

Now consider the initial configuration f^0 in which all assignments in $\mathcal{A}$ are active, and consider an arbitrary disparity α. f^0 is clearly a unique configuration. There is an obvious one-to-one correspondence between the configurations f within a single α-expansion of f^0 and the subsets $\mathcal{U}$ of $\mathcal{V}$. Let $f(\mathcal{U})$ be the configuration, corresponding to the subset $\mathcal{U} \subset \mathcal{V}$.

It's easy to see that the data cost in the energy of the configuration $f(\mathcal{U})$ is zero, the occlusion cost is $\frac{1}{2} \cdot 2(|\mathcal{V}| - |\mathcal{U}|) = |\mathcal{V}| - |\mathcal{U}|$ and the smoothness cost is zero if the subset $|\mathcal{U}|$ is independent, and at least C otherwise. Thus, minimizing the energy in the expansion problem is equivalent to maximizing the cardinality of $\mathcal{U}$ among the independent subsets of $\mathcal{V}$.

Appendix B: Minimizing E with the smoothness term (4) is NP-hard

It is shown in [18] that the following problem, referred to as Potts energy minimization, is NP-hard. We are given as input a set of pixels $\mathcal{S}$ with a neighborhood system $\mathcal{N} \subset \mathcal{S} \times \mathcal{S}$, and a set of label values $\mathcal{V}$ and a non-negative function $D : \mathcal{S} \times \mathcal{V} \mapsto \Re^{+}$. We seek the labeling $f : \mathcal{S} \mapsto \mathcal{V}$ that minimizes

$$E_P(f) = \sum_{p \in \mathcal{P}} D(p, f(p)) + \sum_{\{p,q\} \in \mathcal{N}} T(f(p) \neq f(q)). \tag{14}$$

We now sketch a proof that an arbitrary instance of the Potts energy minimization problem can be encoded as a problem minimizing the energy E defined in equation 2 with the smoothness term (4). This shows that the problem of minimizing E is also NP-hard.

We start with a Potts energy minimization problem consisting of $\mathcal{S}$, $\mathcal{V}$, $\mathcal{N}$ and $\mathcal{D}$. We will create a new instance of our energy minimization problem as follows. The left image $\mathcal{L}$ will be $\mathcal{S}$. For each label in $\mathcal{V}$ we will create a disparity, such that the difference between any pair of disparities is greater than twice the width of $\mathcal{L}$. Obviously, the right image $\mathcal{R}$ will be very large; for every pixel $p \in \mathcal{S}$ and every disparity, there will be a unique pixel in $\mathcal{R}$. The set $\mathcal{A}$ will be pairs of pixels such that there is a disparity where they correspond. Note that two different pixels in $\mathcal{L}$ cannot be mapped to one pixel in $\mathcal{R}$. The penalty for occlusions C_p will be K for $p \in \mathcal{P}$, where

K is a sufficiently large number to ensure that no pixel in $\mathcal{P}$ is occluded in the solution that minimizes the energy E. The neighborhood system will be the Potts model neighborhood system $\mathcal{N}$ extended in the obvious way. The penalty for discontinuities is $V_{a1,a2} = \frac{1}{2}$.

It is now obvious that the global minimum solution to our energy minimization problem will effectively assign a label in $\mathcal{V}$ to each pixel in $\mathcal{S}$. The energy E will be equal to E_P plus a constant, so this global minimum would solve the NP-hard Potts energy minimization problem.

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] P.N. Belhumeur and D. Mumford. A Bayesian treatment of the stereo correspondence problem using half-occluded regions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 506–512, 1992. Revised version appears in *IJCV*.

[3] Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, April 1998.

[4] A.F. Bobick and S.S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):1–20, September 1999.

[5] Robert C. Bolles and John Woodfill. Spatiotemporal consistency checking of passive range data. In *International Symposium on Robotics Research*, 1993. Pittsburg, PA.

[6] Yuri Boykov, Olga Veksler, and Ramin Zabih. Markov random fields with efficient approximations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.

[7] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *International Conference on Computer Vision*, pages 377–384, September 1999.

[8] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[9] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *International Journal of Computer Vision*, 14(3):211–226, April 1995.

[10] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[11] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.

[12] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *European Conference on Computer Vision*, pages 232–248, 1998.

[13] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 125–131, 1998.

[14] Tomaso Poggio, Vincent Torre, and Christof Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[15] R. Potts. Some generalized order-disorder transformation. *Proceedings of the Cambridge Philosophical Society*, 48:106–109, 1952.

[16] S. Roy. Stereo without epipolar lines: A maximum flow formulation. *International Journal of Computer Vision*, 1(2):1–15, 1999.

[17] Rick Szeliski and Ramin Zabih. An experimental comparison of stereo algorithms. In *IEEE Workshop on Vision Algorithms*, September 1999. To appear in *LNCS*.

[18] Olga Veksler. *Efficient Graph-based Energy Minimization Methods in Computer Vision*. PhD thesis, Cornell University, August 1999. Available as technical report CUCS-TR-2000-1787.

[19] C. Lawrence Zitnick and Takeo Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684, July 2000. Earlier version appears as technical report CMU-RI-TR-98-30.