Graph Cut Algorithms for Binocular Stereo with Occlusions

Vladimir Kolmogorov, Ramin Zabih

ABSTRACT Most binocular stereo algorithms assume that all scene elements are visible from both cameras. Scene elements that are visible from only one camera, known as occlusions, pose an important challenge for stereo. Occlusions are important for segmentation, because they appear near discontinuities. However, stereo algorithms tend to ignore occlusions because of their difficulty. One reason is that occlusions require the input images to be treated symmetrically, which complicates the problem formulation. Worse, certain depth maps imply physically impossible scene configurations, and must be excluded from the output. In this chapter we approach the problem of binocular stereo with occlusions from an energy minimization viewpoint. We begin by reviewing traditional stereo methods that do not handle occlusions. If occlusions are ignored, it is easy to formulate the stereo problem as a pixel labeling problem, which leads to an energy function that is common in early vision. This kind of energy function can be minimized using graph cuts, which is a combinatorial optimization technique that has proven to be very effective for low-level vision problems. Motivated by this, we have designed two graph cut stereo algorithms that are designed to handle occlusions. These algorithms produce promising experimental results on real data with ground truth.

1 Traditional stereo methods

Computing stereo depth is a traditional problem in computer vision, and has been the focus of a great deal of work (see [5, 17] for recent surveys). Given a pair of images taken at the same time, two pixels are said to correspond if they show the same scene element. The goal of stereo is to compute correspondences between pixels, which then determines depth. The binocular stereo problem is typically formulated as follows:

> For every pixel in one image, find the corresponding pixel in the other image.

We will refer to this as the *traditional stereo problem*.

The problem formulation above has many advantages. It easily fits within a class of problems that arise in early vision called *pixel labeling problems*,

where the goal is to assign each pixel $p = (p_x, p_y) \in \mathcal{P}$ a label from some set $\mathcal{L}$. The label set $\mathcal{L}$ depends upon the particular problem; for example, in image denoising, $\mathcal{L}$ is intensities. In stereo, $\mathcal{L}$ consists of disparities.

Pixel labeling problems have been widely studied in computer vision. The problem is naturally formulated in terms of energy minimization, where the goal is to find the labeling $f = (f_1, \ldots, f_p, \ldots, f_{|\mathcal{P}|})$ that minimizes

$$E(f) = \sum_p D_p(f_p) + \sum_{\{p,q\} \in \mathcal{N}} V(f_p, f_q). \tag{1.1}$$

Here D_p is the penalty for assigning a label to the pixel p; $\mathcal{N}$ is a set of pairs of adjacent pixels, representing a neighborhood system; and V is the penalty for assigning a pair of labels to adjacent pixels. The first term of equation 1.1 gives a data cost for f, which requires f to respect the observed data, while the second term imposes spatial smoothness. Note that this energy function has an elegant connection to the probabilistic framework provided by Markov Random Fields [14], where the first term comes from the likelihood and the second comes from the prior.

The traditional stereo problem can be easily formulated as a pixel labeling problem. We will assign the label f_p to the pixel p when the pixel p in one image I corresponds to the pixel $p+f_p$ in the other image I'. (Note that the set $\mathcal{P}$ consists of pixels in I.) The *matching penalty* D_p will enforce photoconsistency, which is the tendency of corresponding pixels to have similar intensities. The natural form of D_p is $D_p(f_p) = ||I(p) - I'(p + f_p)||^2$.

The smoothness penalty V will depend on what kind of scene geometry we expect. If V gives too large a penalty for very different f_p, f_q, the solution will tend to oversmooth. With fronto-parallel scenes, the natural choice is $V(f_p, f_q) = \lambda \cdot T[f_p \neq f_q]$, where the indicator function $T[\cdot]$ is 1 if its argument is true and otherwise 0. This choice of V is referred to as the Potts model. There are also more complex forms of V that naturally handle slanted or curved surfaces [2, 4, 15] (surprisingly, these often rely on the Potts model).

The terms D and V can be easily visualized as tables, which are $|\mathcal{L}| \times 1$ or $|\mathcal{L}| \times |\mathcal{L}|$, respectively. For stereo with the Potts model, they are

$$D_p = \begin{array}{|c|} \hline (I(p_x, p_y) - I'(p_x, p_y))^2 \\ \hline (I(p_x, p_y) - I'(p_x - 1, p_y))^2 \\ \hline (I(p_x, p_y) - I'(p_x - 2, p_y))^2 \\ \hline \vdots \\ \hline \end{array} \qquad V = \begin{array}{|c|c|c|c|} \hline 0 & \lambda & \cdots & \lambda \\ \hline \lambda & 0 & \cdots & \lambda \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \lambda & \lambda & \cdots & 0 \\ \hline \end{array}$$

This visualization will prove useful when we describe how to minimize the energy function.

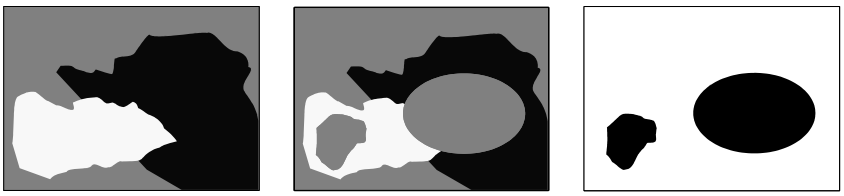


FIGURE 1. Expansion move example. The input labeling is shown at left. An expansion move is shown in the middle, and the corresponding binary labeling is shown at right.

1.1 Energy minimization via graph cuts

A major advantage of pixel labeling problems is that they can now be rapidly solved by powerful optimization algorithms such as graph cuts [4, 6]. If the label set $\mathcal{L}$ consists of contiguous integers and if V is a convex function of $f_p - f_q$, then the global minimum of E can be rapidly computed in a single graph cut [6]. However, if V is convex it will give a large penalty for very different f_p, f_q, and hence will oversmooth. Any class of smoothness terms that includes the Potts model is NP-hard to minimize [4], so a good local minimum is the best that we can hope to achieve.

If V is a metric on labels, then it is possible to efficiently minimize E using the expansion move algorithm. The Potts model is a metric, as are some other popular choices of V that do not oversmooth [4]. The expansion move algorithm computes a strong local minimum, in a sense that we will describe with more precision shortly. Given a label α and a labeling f, another labeling f' is defined to be an α-expansion move from f if for every pixel p

$$f'(p) \neq f(p) \implies f'(p) = \alpha.$$

Intuitively, f' is obtained from f by assigning the label α to an arbitrary set of pixels. An example of an expansion move is shown in figure 1, with f at the left and f' in the middle.

The expansion move algorithm cycles through the labels in some order (fixed or random). For a particular label α, it computes the lowest energy expansion move from the current labeling, and moves to that labeling if its energy is lower. This is obviously a greedy algorithm, and terminates with a labeling that is a local minimum with respect to expansion moves. More precisely, when it terminates with a labeling $\hat{f}$ there is no α-expansion move from $\hat{f}$ whose energy is lower than $E(\hat{f})$, for any label α.

The number of expansion moves from a given labeling is $\mathcal{O}(|\mathcal{L}| \cdot 2^{|\mathcal{P}|})$ (recall that $|\mathcal{P}|$ is the number of pixels). It is possible to prove that the energy of a local minimum with respect to expansion moves lies within a fixed multiplicative factor of the energy of the global minimum. The factor is at least 2, and depends on the exact form of V (see [4] for details).

The key challenge in the expansion move algorithm lies in solving the

following subproblem: given a labeling f and a label α, find the lowest energy α-expansion move from f. In an expansion move, each pixel p has two options: it can keep its old label f_p, or it can switch to the new label α. As a result, an expansion move can be naturally viewed as a binary image; there is a single bit assigned to each pixel, representing which option that pixel selects in this expansion move. For example, figure 1 shows at right the binary image corresponding to the expansion move at center.

We can thus view the problem of finding the lowest energy expansion move as an energy minimization problem over binary images. To formalize this, consider a binary image $\mathbf{x} = \{ x_p \mid p \in \mathcal{P} \}$. The labeling associated with $\mathbf{x}$, given an initial labeling f and a label α, will be α at pixels where $\mathbf{x}$ is 1, and the same as f elsewhere. We will write this labeling as $f^\alpha[\mathbf{x}]$. The problem of finding the lowest energy expansion move is to find the $\mathbf{x}$ that minimizes $E(f^\alpha[\mathbf{x}])$, given f and α.

We can now rewrite the energy E as a new energy function $\mathcal{E}(\mathbf{x})$, where $\mathcal{E}(\mathbf{x}) = E(f^\alpha[\mathbf{x}])$. The new energy function is defined on binary images, and is given by

$$\mathcal{E}(\mathbf{x}) = \sum_p \mathcal{E}_p(x_p) + \sum_{p,q} \mathcal{E}_{p,q}(x_p, x_q)$$

Just as before, the two terms can be visualized as tables, where

$$\mathcal{E}_p = \begin{array}{|c|}\hline D_p(f_p) \\ \hline D_p(\alpha) \\ \hline \end{array} \qquad \mathcal{E}_{p,q} = \begin{array}{|c|c|}\hline V(f_p, f_q) & V(f_p, \alpha) \\ \hline V(\alpha, f_q) & V(\alpha, \alpha) \\ \hline \end{array}$$

The problem of minimizing $\mathcal{E}(\mathbf{x})$ can be solved exactly with a single graph cut as long as $\mathcal{E}_{p,q}$ has a property called *regularity*, introduced in [11]. $\mathcal{E}_{p,q}$ is regular if the sum of its diagonal elements is less than or equal to the sum of its off-diagonal elements; so a sufficient condition is

$$V(\alpha, \alpha) + V(l, l') \leq V(l, \alpha) + V(\alpha, l') \tag{1.2}$$

for any labels l, l', α. As long as this condition is met, the general-purpose construction given in [11] can be used to minimize $\mathcal{E}$, and hence to find the lowest energy expansion move. Note that if V is a metric, it clearly satisfies this condition since $V(\alpha, \alpha) = 0$ and so equation 1.2 is just the triangle inequality.

In summary, the traditional stereo problem is a pixel labeling problem. With the appopriate choices of D_p and V it can be formulated as an energy minimization problem. When V is a metric, a strong local minimum can be computed using the expansion move algorithm. This stereo algorithm, due to [4], yields very good experimental results. For example, the majority of the top-ranked methods on the Middlebury stereo database rely on graph cuts [17].

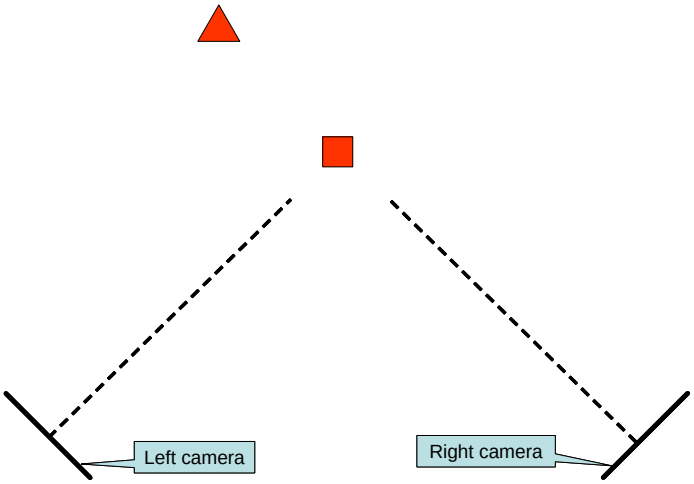


FIGURE 2. It is impossible for the left camera to see the square and the right camera to see the triangle at the same time.

2 Stereo with occlusions

The traditional stereo problem formulation, however, has some serious disadvantages. First, note that the problem formulation treats the input images asymmetrically, which is unnatural. The pixels to be labeled $\mathcal{P}$ come from the primary image I, while I' appears only in the data term D_p. Second, by assigning every pixel in I a label, we assume that every scene element is visible in both images. This cannot be true if the scene has more than one depth.[1]

Worst of all, however, certain labelings f imply physically impossible 3D scenes, and hence must be excluded from consideration. This results from the geometry of the imaging process. An example showing this constraint is shown in figure 2. A general-purpose pixel labeling algorithm will almost invariably generate solutions that violate these geometric constraints. For binocular stereo, these geometric constraints center on occlusions, which are scene elements that are only visible from one camera.

In this chapter we describe two binocular stereo algorithms that handle occlusions. We take an energy minimization approach, and rely on the expansion move algorithm to minimize the energy. One key challenge is that graph cuts perform *unconstrained* energy minimization [11], while binocular stereo with occlusions requires a *constrained* energy minimization algo-

[1]While it is possible to augment the label set by adding a label that means "this pixel is occluded", this approach does not address the other difficulties of the traditional problem formulation.

rithm.

The energy minimization approach to binocular stereo with occlusions consists of following three steps:

- Pick a representation for the problem. In other words, we need to choose the space of valid (physically possible) configurations $\mathcal{C}_{valid}$ and define the correspondence between configurations and real scenes.
- Design an energy function $E : \mathcal{C}_{valid} \rightarrow \mathbb{R}$ that captures the desired properties of a solution.
- Develop an algorithm for minimizing this energy.

Note that these steps are strongly interconnected. With a poor choice of representation, it may be hard or impossible to impose the correct problem constraints. Even if an energy function does captures all the desirable properties, computing a good minimum may be computationally intractable. We will get an effective algorithm only if all three issues are properly addressed.

Ideally, a representation for the stereo problem should have the following properties: for a given configuration it should be easy to determine

(P1) whether it is valid or not (i.e. whether there exists a real scene corresponding to this configuration); and

(P2) what pixels in the left and in the right image correspond to each other. This is crucial since photoconsistency should only be imposed between corresponding pixels.

There are two obvious types of representations for stereo: voxel-style representations, and representations based on labeling pixels. Voxel-style representations rely on an explicit representation of the 3D space that the scene may occupy. Such representations have been used in many approaches, including voxel coloring [18], space carving [13] and silhouette intersection [16]. Pixel labeling approaches include all the standard stereo methods, such as those surveyed in [5, 17].

2.1 Notation

We will redefine $\mathcal{P}$ to now be the set of pixels in the left and in the right images (the previous definition was asymmetric). Let $\mathcal{V}$ be the set of (unordered) pairs of pixels that may potentially correspond. For simplicity we assume that images are rectified; then we have

$$\mathcal{V} = \{\langle p, q\rangle \mid p_y = q_y \text{ and } q_x - p_x \in \mathcal{L}\}$$

where $\mathcal{L}$ is the set of possible disparities: $\mathcal{L} = \{0, -1, \ldots, -d_{max}\}$. (We assume that disparities lie in some limited range, so each pixel in the left image can potentially correspond to one of $|\mathcal{L}|$ possible pixels in the right

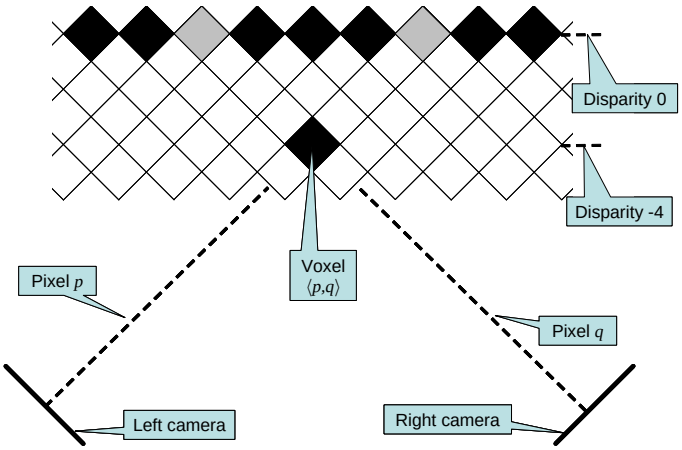


FIGURE 3. Voxel labeling algorithm. Voxels are given a binary label (active or not); dark shaded voxels are labeled as active. The disparity of the voxel $\langle p, q\rangle$ is $d(\langle p, q\rangle) = -4$. To simplify the drawing, orthographic projection is assumed. Note that the two gray-shaded voxels cannot be active if $\langle p, q\rangle$ is active.

image, and vice versa). We call a pair $v = \langle p, q\rangle \in \mathcal{V}$ a *voxel*. Its disparity is denoted as $d(v)$ (i.e. $d(v) = q_x - p_x \in \mathcal{L}$).

Note that each voxel $v \in \mathcal{V}$ corresponds to a point in 3D space, as shown in figure 3. The disparity $d(v)$ directly depends on the depth of this point, i.e. its distance to the cameras. If the cameras are parallel then $-d(v)$ is inversely proportional to the depth. In a more general situation the relationship can be more complicated. In this chapter we assume that disparity is a monotonically increasing function of the depth. In other words, the farther a point from the cameras the larger the disparity.

For a voxel $v = \langle p, q\rangle$ we can compute the matching penalty $M(v)$ describing how photoconsistent the intensity of pixel p is with the intensity of pixel q. The simplest function is the squared difference of intensities: $M(v) = ||I(p) - I'(q)||^2$; however, more elaborate functions (for example, [1]) tend to give better results.

3 Voxel labeling algorithm

Our first approach, which first appeared in [9], is directly inspired by property P2. A configuration will just be a labeling $g : \mathcal{V} \to \{0, 1\}$ such that $g(v)$ is 1 if the pixels p and q in voxel $v = \langle p, q\rangle$ correspond to each other, and 0 otherwise. In other words $g(v) = 1$ if and only if the the 3D point corresponding to voxel v is present in the scene and is visible from both

cameras. If this is the case we will say that v is *active*.

The set of all configurations is $\mathcal{C} = \{0,1\}^{\mathcal{V}}$. However, not all configurations in $\mathcal{C}$ are valid. Some of them violate the *uniqueness* constraint which says that a pixel in one image can correspond to at most one pixel in the other image. Let us define the set $\mathcal{C}_{valid}$ as follows: the configuration $g \in \mathcal{C}$ is valid if for any two distinct voxels v, v' involving the same pixel (i.e. $v = \langle p, q\rangle$, $v' = \langle p, q'\rangle$ with $q \neq q'$) at least one of them has label 0: $g(v) = 0$ or $g(v') = 0$.

Now let us define the energy we will minimize. It has three terms: data, occlusion and smoothness:

$$E(g) = E_{data}(g) + E_{occ}(g) + E_{smooth}(g) \tag{1.3}$$

The data term will be $E_{data}(g) = \sum_{v \in \mathcal{V}} g(v) \cdot M(v)$. Note that this sum contains matching penalties only for voxels v which are active in configuration g (i.e. $g(v) = 1$). The term $E_{occ}(g)$ penalizes occlusions: it is equal to $C_{occ} \cdot |\mathcal{P}_{occ}(g)|$ where C_{occ} is the penalty for an occlusion and $\mathcal{P}_{occ}(g)$ is the set of pixels occluded in configuration g (i.e. pixels p such that $g(v) = 0$ for all voxels $v = \langle p, q\rangle \in \mathcal{V}$).

The smoothness term involves a notion of neighborhood; we assume that there is a neighborhood system on voxels

$$\mathcal{N}_{\mathcal{V}} \subset \{\{v, v'\} \mid v, v' \in \mathcal{V}\}.$$

We require that for every pair $\{v, v'\} \in \mathcal{N}_{\mathcal{V}}$ the disparities of voxels v and v' are the same: $d(v) = d(v')$. For example, we can specify $\mathcal{N}_{\mathcal{V}}$ as follows: voxels $\langle p, q\rangle$,$\langle p', q'\rangle$ with the same disparity are neighbors if pixels p, p' in the left image are 4-neighbors. Now the smoothness term can be written:

$$E_{smooth}(g) = \sum_{\{v,v'\} \in \mathcal{N}_{\mathcal{V}}} \lambda \cdot T[g(v) \neq g(v')].$$

To summarize, the voxel labeling stereo algorithm solves the constrained minimization problem:

$$g^* = \arg \min_{g \in \mathcal{C}_{valid}} E(g), \tag{1.4}$$

where $E(g)$ is defined in equation 1.3.

4 Pixel labeling algorithm

The representation discussed above might seem natural for stereo correspondence problem since it allows to identify corresponding pixels easily. However, it has several drawbacks. First, the smoothness term involved is rather restrictive — basically, it is the Potts model on voxels (see [12] for

more details). Second, the set $\mathcal{C}_{valid}$ contains configurations which do not correspond to any physical scene. Consider, for example, the configuration g with $g(a) = 0$ for every voxel $v \in \mathcal{V}$. Every pixel is occluded in this configuration; thus, the configuration contains "holes". From a practical point of view, we can ensure that we will not get such a configuration by setting penalty for occlusion to a sufficiently large value.

We now describe a different approach, first published in [10], which uses a representation proposed by [7]. We know that each pixel sees some element of the scene (even though this element may not be seen from the other camera). Our goal will be to compute the depth of this pixel (or, rather, its disparity). Thus, a configuration is a mapping $f : \mathcal{P} \to \mathcal{L}$. The set of all configurations is $\mathcal{C} = \mathcal{L}^{\mathcal{P}}$.

As in the previous case, not all configurations are valid. Formally, the configuration $f \in \mathcal{C}$ is valid if for every voxel $v = \langle p, q \rangle$ the following property holds: if $f(p) = d(v)$ then $f(q) \leq d(v)$. This can be understood intuitively in terms of figure 2; if the left camera sees the square, the right camera cannot see any scene element that is behind the square.

Our energy function will be

$$E(f) = E_{data}(f) + E_{smooth}(f). \tag{1.5}$$

Similar to the previous case, we would like the data term to be a sum only over *active* voxels. Let us discuss how we can identify such voxels in this representation. The voxel $v = \langle p, q \rangle$ is active if the corresponding 3D point is present in the scene and is visible from both cameras. This means that $f(p) = f(q) = d(v)$. This in turn motivates the following data term:

$$E_{data}(f) = \sum_{v=\langle p,q \rangle \in \mathcal{V}} [f(p) = f(q) = d(v)] \cdot D(v)$$

where $D(v)$ measures how similar intensities of pixels p and q are. For technical reasons explained in section 5 we need the term $D(v)$ to be non-positive. We set $D(v) = \min\{M(v) - K, 0\}$ where K is a positive constant.

The smoothness term is very similar to that of tradional stereo problem, except that it is enforced for both images rather than just the left image:

$$E_{smooth}(f) = \sum_{\{p,q\} \in \mathcal{N}} V(f_p, f_q)$$

where V can be, for example, the Potts model: $V(f_p, f_q) = \lambda \cdot T[f_p \neq f_q]$.

We thus obtain the following constrained minimization problem:

$$f^* = \arg \min_{f \in \mathcal{C}_{valid}} E(f), \tag{1.6}$$

where $E(f)$ is defined in equation 1.5.

5 Minimizing the energy

In this section we sketch how we solve the constrained minimization problems given in equations 1.4 and 1.6. First, we convert our constrained minimization problems into unconstrained ones. We add a hard constraint term E_{valid} which is zero if a configuration is valid, and infinite otherwise. In the case of the pixel labeling algorithm, for example, the energy becomes

$$E(f) = E_{data}(f) + E_{smooth}(f) + E_{valid}(f).$$

All terms of this energy (including E_{valid}) can be written as a sum over pairs of pixels. In other words, the energy has the same functional form as in equation 1.1, only the neighborhood system $\mathcal{N}$ is different and terms V are replaced by some other functions. Moreover, the representation of our problem resembles that of traditional stereo problem (section 1). As in traditional stereo, our goal is to assign disparities to pixels; the only change in representation is to consider pixels in both images. Thus, it is easy to adapt the expansion move algorithm described in section 1 to our minimization problem. We just need to ensure that for each α-expansion the corresponding binary energy function is regular. We show in [10] that this condition holds assuming that terms $D(v)$ are non-positive.

In order to apply the expansion move algorithm to the voxel labeling problem, we need to modify the definition of α-expansion. Indeed, the definition given in section 1 applies to multi-label variables, while our problem has binary labels. We say that configuration g' is within a single α-expansion move from configuration g if voxels which are inactive in g and whose disparity is different from α are also inactive in g'. Then for every valid configuration g and disparity α it is possible to compute an optimal α-expansion move using graph cuts (see [9] for details).

6 Experimental results

6.1 Implementational details

Expansion move algorithm We selected disparities $\alpha \in \mathcal{L}$ in random order, and kept this order for all iterations. We performed three iterations. (The number of iterations until convergence was at most five but the result was practically the same). The voxel labeling algorithm was initialized with a configuration where every voxel was inactive; the pixel labeling algorithm was initialized with all pixels having disparity zero.

Matching penalty For our matching penalty M we made use of the method of [1] to handle sampling artifacts, with a slight variation: we compute intensity intervals for each band (R,G,B) using four neighbors, and then take the average data penalty. (We used color images; results for grayscale images are slightly worse).

Smoothness terms We used a Potts model for both algorithms (in one case this is the Potts model on voxels, while in the other the Potts model on pixels). This model is controlled by one parameter characterizing the penalty for a pair of neighboring voxels or pixels. This parameter, however, can depend on the pair. We can use this property to discourage discontinuities between adjacent pixels with very similar intensities. This trick is referred to as "static cues" in [4] and is quite useful for stereo.

For the pixel labeling algorithm we set

$$V_{p,p'}(f_p, f_{p'}) = \lambda_{p,p'} \cdot T[f_p \neq f_{p'}]$$

where $\lambda_{p,p'}$ was implemented as the following empirically selected decreasing function of $\Delta I(p, p')$ (the L_∞ norm of the intensity difference between p and p'):

$$\lambda_{p,p'} = \begin{cases} 3\lambda & \text{if } \Delta I(p,p') < 5, \\ \lambda & \text{otherwise.} \end{cases}$$

For the voxel labeling algorithm we used a similar expression:

$$\lambda_{v,v'} = \begin{cases} 3\lambda & \text{if } \max(\Delta I(p,p'), \Delta I(q,q')) < 8, \\ \lambda & \text{otherwise,} \end{cases}$$

where $v = \langle p, q \rangle$, $v' = \langle p', q' \rangle$ and p and p' are pixels in the same image, as well as q and q'.

Choice of parameters The energy function for the voxel labeling algorithm as defined above depends on two numbers: occlusion penalty C_{occ} and smoothness interaction strength λ. Similarly, the pixel labeling algorithm depends on the parameters K and λ. These parameters should be tuned for for different datasets to reflect our prior knowledge about the scene geometry, amount of noise in the images and other factors. Selecting the parameters automatically, however, is a very challenging task.

We set K in the pixel labeling algorithm using a simple heuristic which tries to estimate the amount of noise in the images. Details are given in [8]. It can be shown [8, 12] that $K/2$ approximately corresponds to the occlusion penalty, so for the voxel labeling algorithm we set $C_{occ} = K/2$. Finally, the parameter λ was chosen to be proportional to K: $\lambda = K/5$.

6.2 Algorithm performance

We have compared three algorithms: our voxel and pixel labeling algorithms with occlusions ("[KZ '01]" and "[KZ '02]") and a traditional stereo algorithm proposed in [4] ("[BVZ]"). The latter technique was found to be the best algorithm for stereo according to [19]. In addition, we tested the algorithms in two modes: with reporting occlusions (some of the pixels in the left image are marked as occluded) and without reporting occlusions (all pixels in the left image are labeled with some disparity).

Determining occluded areas in the voxel and pixel labeling algorithms is easy since they output what pixels correspond to each other. The information produced by [BVZ], however, is not sufficient to determine where occlusions are. To produce occlusions, we have augmented the algorithm: we introduced a new label "occluded" with some fixed penalty.

Note that our voxel labeling algorithm does not produce depths for all pixels. We have filled occluded regions using some postprocessing: we have assigned to occluded pixels the depth label of the closest non-occluded left neighbor lying in the same scanline.

We primarily experimented with images from [17]; output is shown in figures 4– 6. The running times below were obtained on 450MHz UltraSPARC II processor. We used the max flow algorithm of [3], which is specifically designed for the kinds of graphs that arise in vision.

stereo pair	image size	number of labels	running times [KZ '01]	[KZ '02]	[BVZ]
Tsukuba	384 x 288	16	69 secs	80 secs	35 secs
Sawtooth	434 x 380	20	115 secs	141 secs	66 secs
Venus	434 x 383	22	145 secs	159 secs	85 secs

First we evaluated the three algorithms in the mode without reporting occlusions. Error statistics using the ground truth from [17] are as follows:

stereo pair	[KZ '01]	[KZ '02]	[BVZ]
Tsukuba	5.82 (1.18)	5.91 (1.86)	7.17 (1.93)
Sawtooth	12.13 (0.71)	11.77 (0.67)	11.86 (0.62)
Venus	15.40 (1.07)	13.19 (0.69)	16.90 (0.75)

We determined the percentage of the pixels where the algorithm did not compute the correct disparity ("errors" — the first number), or a disparity within ± 1 of the correct disparity ("gross errors" — the second number). We counted only pixels that are not occluded according to the ground truth since depth labels of such pixels cannot be determined from the photoconsistency constraint.

We have also computed error statistics for the Tsukuba stereo pair in the mode with reporting occlusions.

algorithm	Errors	Gross errors	False negatives	False positives
[KZ '01]	6.56%	2.17%	41.33%	1.33%
[KZ '02]	6.51%	2.66%	44.16%	1.03%
[BVZ]	7.28%	2.14%	77.59%	0.62%

The first two columns count only pixels that are not occluded according to the ground truth. We considered labeling a pixel as occluded to be a gross error. The last two columns show error rates for occlusions.

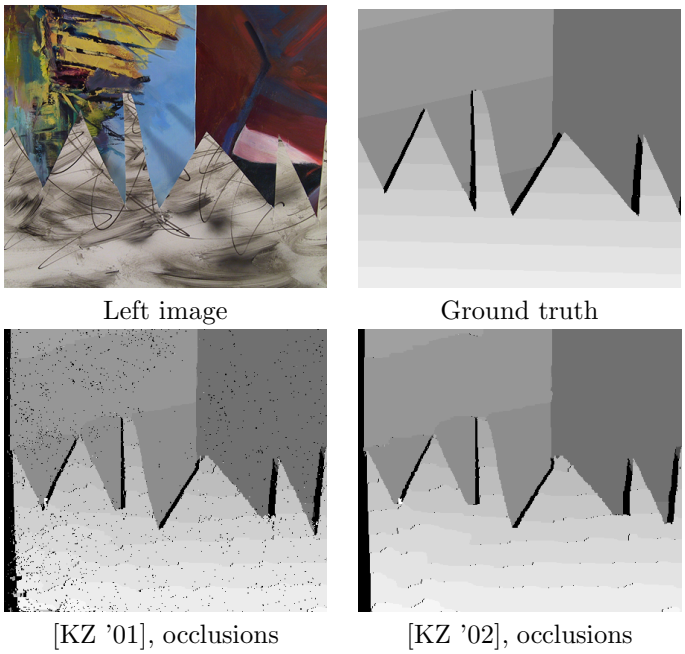


FIGURE 4. Sawtooth results (occlusions are shown in black).

7 Conclusions

We have presented two stereo algorithms that handle occlusions. The pixel labeling algorithm can be viewed as an improvement over the voxel labeling algorithm for two reasons. First, unlike voxel labeling, pixel labeling explicitly prohibits "holes" in the scene. In other words, it takes into account the fact that for any real scene the layer with disparity 0 (corresponding to the plane at infinity) is filled. Second, our pixel labeling method allows not only Potts interactions, but other useful smoothness terms (for example, truncated linear terms).

The major limitation of our approach lies in its bias towards fronto-parallel surfaces. With a sloped surface, our methods yield occlusions at discontinuities resulting from discretizing disparities. These occlusions are treated in the same way as real occlusions at object boundaries. Note that in the pixel labeling algorithm the problem can be alleviated by using a truncated linear smoothness term instead of Potts model.

It is possible to extend our algorithms to handle multiple cameras [8, 10, 12]. However, no scene point can lie inside the convex hull of the camera centers. This is the same class of camera configurations where voxel coloring [18] can be used, and includes many situations of practical interest.

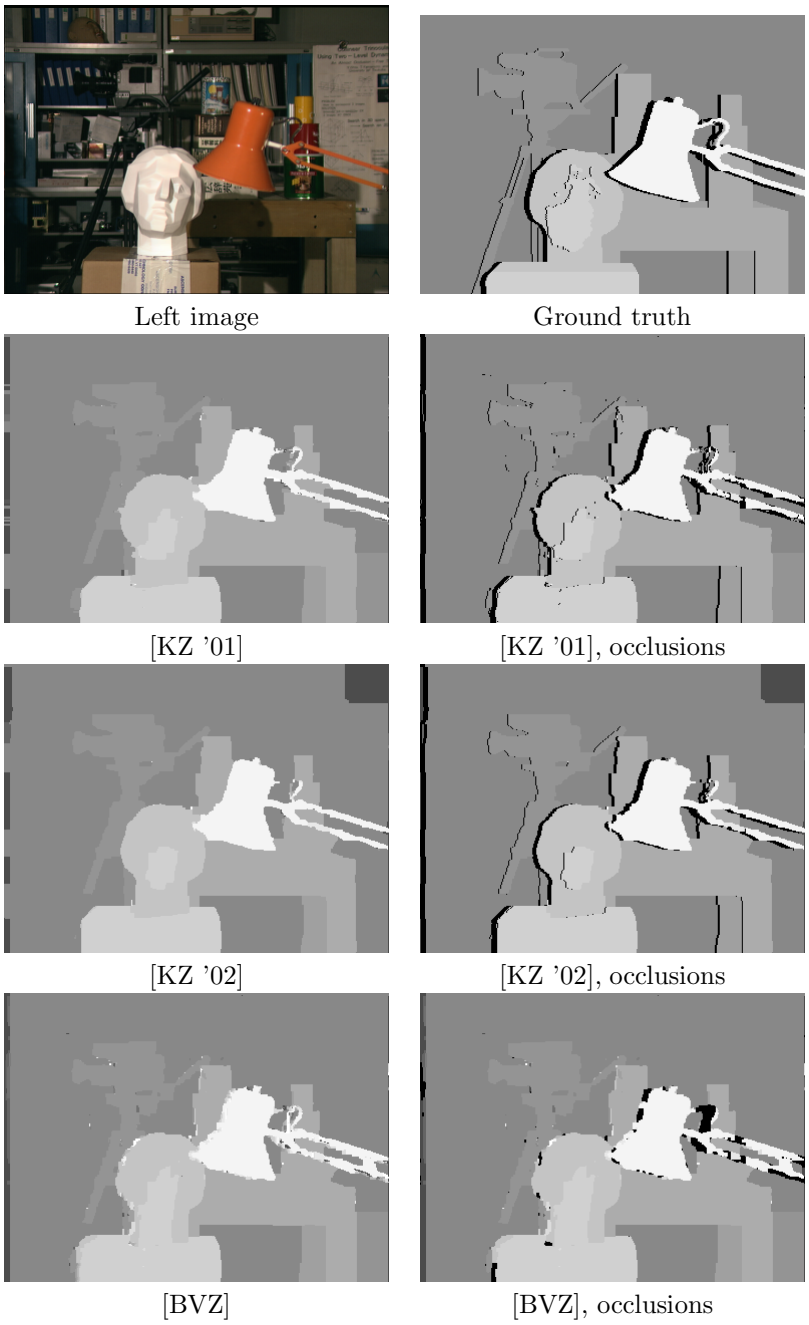
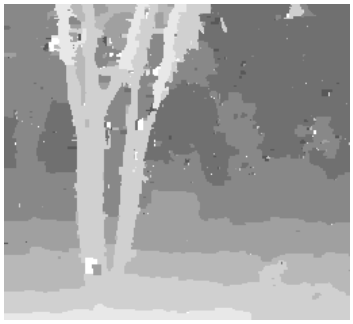
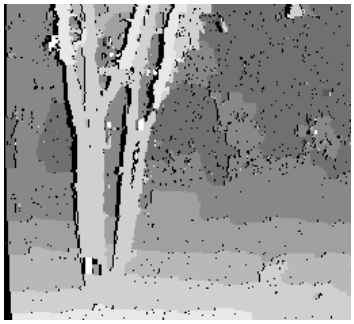


FIGURE 5. Tsukuba results (occlusions are shown in black).

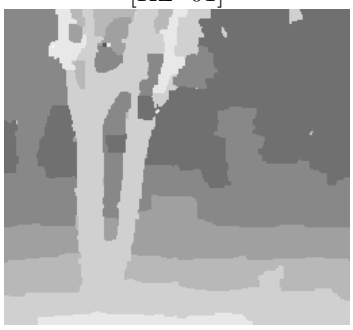
Left image



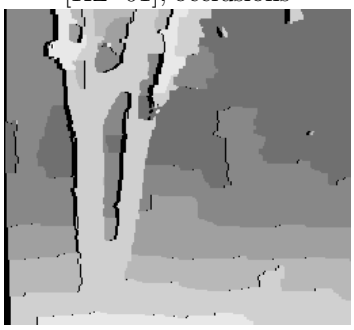
[KZ '01]



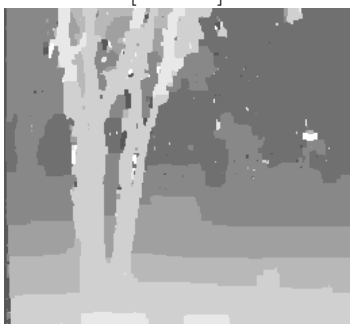
[KZ '01], occlusions



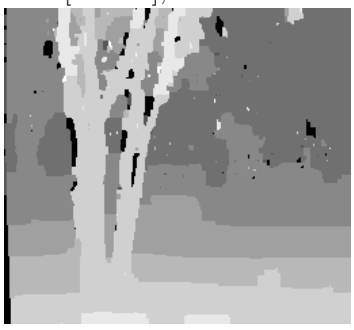
[KZ '02]



[KZ '02], occlusions



[BVZ]



[BVZ], occlusions

FIGURE 6. Tree image results (occlusions are shown in black).

8 References

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

[2] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *International Conference on Computer Vision*, pages 489–495, 1999.

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, Nov. 2001.

[5] M. Brown, D. Burschka, and G. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, August 2003.

[6] H. Ishikawa. Exact optimization for Markov Random Fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, Oct. 2003.

[7] S. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001. Expanded version available as MSR-TR-2001-80.

[8] V. Kolmogorov. *Graph Based Algorithms for Scene Reconstruction from Two or More Views*. PhD thesis, Cornell University, Sept. 2003.

[9] V. Kolmogorov and R. Zabih. Visual correspondence with occlusions using graph cuts. In *International Conference on Computer Vision*, pages 508–515, 2001.

[10] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision*, volume 3, pages 82–96, 2002.

[11] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, Feb. 2004.

[12] V. Kolmogorov, R. Zabih, and S. Gortler. Generalized multi-camera scene reconstruction using graph cuts. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, July 2003.

[13] K. Kutulakos and S. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):197–216, July 2000.

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

[15] M. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1073–1078, August 2004.

[16] W. Martin and J. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, March 1983.

[17] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, Apr. 2002.

[18] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):1–23, November 1999.

[19] R. Szeliski and R. Zabih. An experimental comparison of stereo algorithms. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, number 1883 in LNCS, pages 1–19, Corfu, Greece, Sept. 1999. Springer-Verlag.