Multi-camera Scene Reconstruction via Graph Cuts

Vladimir Kolmogorov and Ramin Zabih

Computer Science Department, Cornell University, Ithaca, NY 14853
vnk@cs.cornell.edu, rdz@cs.cornell.edu

Abstract. We address the problem of computing the 3-dimensional shape of an arbitrary scene from a set of images taken at known viewpoints. Multi-camera scene reconstruction is a natural generalization of the stereo matching problem. However, it is much more difficult than stereo, primarily due to the difficulty of reasoning about visibility. In this paper, we take an approach that has yielded excellent results for stereo, namely energy minimization via graph cuts. We first give an energy minimization formulation of the multi-camera scene reconstruction problem. The energy that we minimize treats the input images symmetrically, handles visibility properly, and imposes spatial smoothness while preserving discontinuities. As the energy function is NP-hard to minimize exactly, we give a graph cut algorithm that computes a local minimum in a strong sense. We handle all camera configurations where voxel coloring can be used, which is a large and natural class. Experimental data demonstrates the effectiveness of our approach.

1 Introduction

Reconstructing an object's 3-dimensional shape from a set of cameras is a classic vision problem. In the last few years, it has attracted a great deal of interest, partly due to a number of new applications both in vision and in graphics that require good reconstructions. While the problem can be viewed as a natural generalization of stereo, it is considerably harder. The major reason for this is the difficulty of reasoning about visibility. In stereo matching, most scene elements are visible from both cameras, and it is possible to obtain good results without addressing visibility constraints. In the more general scene reconstruction problem, however, very few scene elements are visible from every camera, so the issue of visibility cannot be ignored.

In this paper, we approach the scene reconstruction problem from the point of view of energy minimization. Energy minimization has several theoretical advantages, but has generally been viewed as too slow for early vision to be practical. Our approach is motivated by some recent work in early vision, where fast energy minimization algorithms have been developed based on graph cuts [6, 7, 12, 14, 20, 21]. These methods give strong experimental results in practice, as documented in two recent evaluations of stereo algorithms using real imagery with dense ground truth [22, 27].

The energy that we minimize has three important properties:

– it treats the input images symmetrically,
– it handles visibility properly, and
– it imposes spatial smoothness while also preserving discontinuities.

We begin with a review of related work. In section 3 we give a precise definition of the problem that we wish to solve, and define the energy that we will minimize. Section 4 describes how we use graph cuts to compute a strong local minimum of this energy. Experimental data is presented in section 5. Some details of the graph cut construction are deferred to the appendix, along with a proof that computing the global minimum of the energy is NP-hard.

2 Related work

The problem of reconstructing a scene from multiple cameras has received a great deal of attention in the last few years. One extensively-explored approach to this problem is voxel occupancy. In voxel occupancy [18, 25] the scene is represented as a set of 3-dimensional voxels, and the task is to label the individual voxels as filled or empty. Voxel occupancy is typically solved using silhouette intersection, usually from multiple cameras but sometimes from a single camera with the object placed on a turntable [8]. It is known that the output of silhouette intersection even without noise is not the actual 3-dimensional shape, but rather an approximation called the visual hull [17].

2.1 Voxel coloring and space carving

Voxel occupancy, however, fails to exploit the consistent appearance of a scene element between different cameras. This constraint, called *photo-consistency*, is obviously quite powerful. Two well-known recent algorithms that have used photo-consistency are voxel coloring [23] and space carving [16].

Voxel coloring makes a single pass through voxel space, first computing the visibility of each voxel and then its color. There is a constraint on the camera geometry, namely that no scene point is allowed to be within the convex hull of the camera centers. As we will see in section 3, our approach handles all the camera configurations where voxel coloring can be used. Space carving is another voxel-oriented approach that uses the photo-consistency constraint to prune away empty voxels from the volume. Space carving has the advantage of allowing arbitrary camera geometry.

One major limitation of voxel coloring and space carving is that they lack a way of imposing spatial coherence. This is particularly problematic because the image data is almost always ambiguous. Another (related) limitation comes from the fact that these methods traverse the volume making “hard” decisions concerning the occupancy of each voxel they analyze. Because the data is ambiguous, such a decision can easily be incorrect, and there is no easy way to undo such a decision later on.

2.2 Energy minimization approaches

Our approach to the scene reconstruction problem is to generalize some recently developed techniques that give strong results for stereo matching. It is well known that stereo, like many problems in early vision, can be elegantly stated in terms of energy minimization [19]. The energy minimization problem has traditionally been solved via simulated annealing [2, 11], which is extremely slow in practice.

In the last few years powerful energy minimization algorithms have been developed based on graph cuts [6, 7, 12, 14, 21]. These methods are fast enough to be practical, and yield quite promising experimental results for stereo [22, 27]. Unlike simulated annealing, graph cut methods cannot be applied to an arbitrary energy function; instead, for each energy function to be minimized, a careful graph construction must be developed. In this paper, instead of building a special purpose graph we will use some recent results [15] that give graph constructions for a quite general class of energy functions.

While energy minimization has been widely used for stereo, only a few papers [13, 20, 24] have used it for scene reconstruction. The energy minimization formalism has several advantages. It allows a clean specification of the problem to be solved, as opposed to the algorithm used to solve it. In addition, energy minimization naturally allows the use of soft constraints, such as spatial coherence. In an energy minimization framework, it is possible to cause ambiguities to be resolved in a manner that leads to a spatially smooth answer. Finally, energy minimization avoids being trapped by early hard decisions.

Although [20] and [24] use energy minimization via graph cuts, their focus is quite different from ours. [20] uses an energy function whose global minimum can be computed efficiently via graph cuts; however, the spatial smoothness term is not discontinuity preserving, and so the results tend to be oversmoothed. Visibility constraints are not used. [24] computes the global minimum of a different energy function as an alternative to silhouette intersection (i.e., to determine voxel occupancy). Their approach does not deal with photoconsistency at all, nor do they reason about visibility.

Our method is perhaps closest to the work of [13], which also relies on graph cuts. They extend the work of [7], which focused on traditional stereo matching, to allow an explicit label for occluded pixels. While the energy function that they use is of a similar general form to ours, they do not treat the input images symmetrically. While we effectively compute a disparity map with respect to each camera, they compute a disparity map only with respect to a single camera.

[26] also proposed the use of disparity maps for each camera, in an energy minimization framework. Their optimization scheme did not rely on graph cuts, and their emphasis was on the two-camera stereo problem. They also focused on handling transparency, which is an issue that we do not address.

Another approach is to formulate the problem in terms of level sets, and to solve a system of partial differential equations using numerical techniques [9]. The major distinction between this approach and our work is that their problem formulation is continuous while ours is discrete. The discrete energy minimization approach is known to lead to strong results for two-camera stereo [22, 27], and is therefore worth investigating for multiple cameras.

In [14] we proposed a (two camera) stereo algorithm that shares a number of properties with the present work. That method treats the input images symmetrically, and is based on graph cuts for energy minimization. It properly handles the limited form of visibility constraint that arises with two cameras (namely, occlusions). The major difference is that [14] is limited to the case of two cameras, while the multi-camera problem is much more difficult.

3 Problem formulation

Suppose we are given n calibrated images of the same scene taken from different viewpoints (or at different moments of time). Let $\mathcal{P}_i$ be the set of pixels in the camera i, and let $\mathcal{P} = \mathcal{P}_1 \cup \ldots \cup \mathcal{P}_n$ be the set of all pixels. A pixel $p \in \mathcal{P}$ corresponds to a ray in 3D-space. Consider the point of the first intersection of this ray with an object in the scene. Our goal is to find the depth of this point for all pixels in all images. Thus, we want to find a labeling $f : \mathcal{P} \rightarrow \mathcal{L}$ where $\mathcal{L}$ is a discrete set of labels corresponding to different depths. In the current implementation of our method, labels correspond to increasing depth from a fixed camera.

A pair $\langle p, l\rangle$ where $p \in \mathcal{P}$, $l \in \mathcal{L}$ corresponds to some point in 3D-space. We will refer to such pairs as *3D-points*.

The limitation of our method is that there must exist a function $\mathcal{D} : R^3 \mapsto R$ such that for all scene points P and Q, P occludes Q in a camera i only if $\mathcal{D}(P) < \mathcal{D}(Q)$. This is exactly the constraint used in voxel coloring [23]. If such a function exists then labels correspond to level sets of this function. In our current implementation, we make a slightly more specific assumption, which is that the cameras must lie in one semiplane looking at the other semiplane. The interpretation of labels will be as follows: each label corresponds to a plane in 3D-space, and a 3D-point $\langle p, l\rangle$ is the intersection of the ray corresponding to the pixel p and the plane l.

Let us introduce the set of interactions I consisting of (unordered) pairs of 3D-points $\langle p_1, l_1\rangle$, $\langle p_2, l_2\rangle$ "close" to each other in 3D-space. We will discuss several criteria for "closeness" later. In general, I can be an arbitrary set of pairs of 3D-points satisfying the following constraint:

– Only 3D-points at the same depth can interact, i.e.if $\{\langle p_1, l_1\rangle, \langle p_2, l_2\rangle\} \in I$ then $l_1 = l_2$.

Now we will define the energy function that we minimize. It will consist of three terms:

$$E(f) = E_{data}(f) + E_{smoothness}(f) + E_{visibility}(f) \tag{1}$$

The data term will impose photo-consistency. It is

$$E_{data}(f) = \sum_{\langle p, f(p)\rangle, \langle q, f(q)\rangle \in I} D(p, q)$$

where $D(p,q)$ is a non-positive value depending on intensities of pixels p and q. It can be, for example,

$$D(p,q) = min\{0, (Intensity(p) - Intensity(q))^2 - K\}$$

for some constant $K > 0$.

Since we minimize the energy, terms $D(p,q)$ that we sum will be small. These terms are required to be non-negative for technical reasons that will be described in section 4.2. Thus, pairs of pixels p, q which come from the same scene point according to the configuration f will have similar intensities, which causes photo-consistency.

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

$$\mathcal{N} \subset \{\{p,q\} \mid p,q \in \mathcal{P}\}$$

This can be the usual 4-neighborhood system: pixels $p = (p_x, p_y)$ and $q = (q_x, q_y)$ are neighbors if they are in the same image and $|p_x - q_x| + |p_y - q_y| = 1$. We will write the smoothness term as

$$E_{smoothess}(f) = \sum_{\{p,q\}\in\mathcal{N}} V_{\{p,q\}}(f(p), f(q))$$

We will require the term $V_{\{p,q\}}$ to be a metric. This imposes smoothness while preserving discontinuities, as long as we pick an appropriate robust metric. For example, we can use the robustified L_1 distance $V(l_1, l_2) = \min(|l_1 - l_2|, K)$ for constant K.

The last term will encode the visibility constraint: it will be zero if this constraint is satisfied, and infinity otherwise. We can write this using another set of interactions I_{vis} which contains pairs of 3D-points violating the visibility constraint:

$$E_{visibility}(f) = \sum_{\langle p,f(p)\rangle, \langle q,f(q)\rangle \in I_{vis}} \infty$$

We require the set I_{vis} to meet following condition:

- Only 3D-points at different depths can interact, i.e. if $\{\langle p_1, l_1\rangle, \langle p_2, l_2\rangle\} \in I_{vis}$ then $l_1 \neq l_2$.

The visibility constraint says that if a 3D-point $\langle p, l\rangle$ is present in a configuration f (i.e. $l = f(p)$) then it "blocks" views from other cameras: if a ray corresponding to a pixel q goes through (or close to) $\langle p, l\rangle$ then its depth is at most l. Again, we need a definition of "closeness". We will use the set I this part of the construction of I_{vis}. The set I_{vis} can then be defined as follows: it will contain all pairs of 3D-points $\langle p, l\rangle$, $\langle q, l'\rangle$ such that $\langle p, l\rangle$ and $\langle q, l\rangle$ interact (i.e. they are in I) and $l' > l$.

An important issue is the choice of the sets of interactions I. This basically defines a discretization. It can consist, for example, of pairs of 3D-points

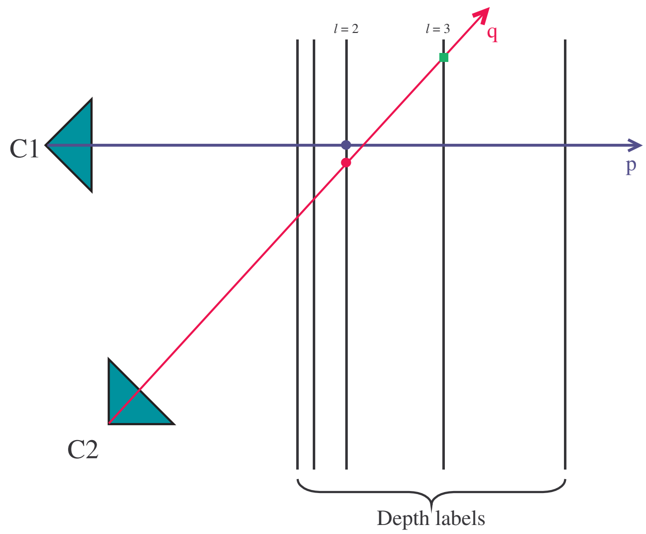


Fig. 1. Example of pixel interactions. There is a photo-consistency constraint between the red round point and the blue round point, both of which are at the same depth ($l = 2$). The red round point blocks camera C2's view of the green square point at depth $l = 3$. Color figures are available in the electronic version of this paper.

$\{\langle p,l\rangle, \langle q,l\rangle\}$ such that the distance in 3D-space between these points is less than some threshold. However, with this definition of closeness the number of interactions per pixel may vary greatly from pixel to pixel. Instead, we chose the following criterion: if p is a pixel in the image i, q is a pixel in the image j and $i < j$ then the 3D-points $\langle p,l\rangle$, $\langle q,l\rangle$ interact if the closest pixel to the projection of $\langle p,l\rangle$ onto the image j is q. Of course, we can include only interactions between certain images, for example, taken by neighboring cameras. Note that it makes sense to include interactions only between *different* cameras.

An example of our problem formulation in action is shown in figure 1. There are two cameras C1 and C2. There are 5 labels shown as black vertical lines. As in our current implementation, labels are distributed by increasing distance from a fixed camera. Two pixels, p from C1 and q from C2 are shown, along with the red round 3D-point $\langle q,2\rangle$ and the blue round 3D-point $\langle p,2\rangle$. These points share the same label, and interact (i.e., $\{\langle p,2\rangle, \langle q,2\rangle\} \in I$). So there is a photoconsistency term between them. The green square point $\langle q,3\rangle$ is at a different label (greater depth), but is behind the red round point. The pair of 3D-points $\{\langle p,2\rangle, \langle q,3\rangle\}$ is in I_{vis}. So if the ray p from camera C1 sees the red round point $\langle p,2\rangle$, the ray q from C2 cannot see the green square point $\langle q,3\rangle$.

We show in the appendix that minimizing our energy is an NP-hard problem. Our approach, therefore, is to constuct an approximation algorithm based on graph cuts that finds a strong local minimum.

4 Graph construction

We now show how to efficiently minimize E among all 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 within a single α-*expansion* of f when for all pixels $p \in \mathcal{P}$ either $f'(p) = f(p)$ or $f'(p) = \alpha$. This notion of an expansion was proposed by [7], and forms the basis for several very effective stereo algorithms [3, 7, 13, 14].

Our algorithm is very straightforward; 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. Except for the problem formulation and the choice of energy function, this algorithm is identical to the methods of [7, 14].

One restriction on the algorithm is that the initial configuration must satisfy the visibility constraint. This will guarantee that all subsequent configurations will satisfy this constraint as well, since we minimize the energy, and configurations that do not satisfy the visibility constraint have infinite energy.

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.

4.1 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 vertices into two sets such that $s \in \mathcal{V}^s$ and $t \in \mathcal{V}^t$. (Note that a cut can also be equivalently defined as the set of edges between the two sets.) 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 [10]. 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.

We will use a result from [15] which says that for energy functions of binary variables of the form

$$E(x_1, \ldots, x_n) = \sum_{i<j} E^{i,j}(x_i, x_j) \tag{2}$$

it is possible to construct a graph for minimizing it if and only if each term $E^{i,j}$ satisfies the following condition:

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0) \tag{3}$$

If these conditions are satisfied then the graph $\mathcal{G}$ is constructed as follows. We add a node v_i for each variable x_i. For each term $E^{i,j}(x_i, x_j)$ we add edges which are given in the appendix.

Every cut on such a graph corresponds to some configuration $x = (x_1, \ldots, x_n)$, and vice versa: if $v_i \in \mathcal{V}^s$ then $x_i = 0$, otherwise $x_i = 1$. Edges on a graph were added in such a way that the cost of any cut is equal to the energy of the corresponding configuration plus a constant. Thus, the minimum cut on $\mathcal{G}$ yields the configuration that minimizes the energy.

4.2 α-expansion

In this section we will show how to convert our energy function into the form of equation 2. Note that it is not necessary to use only terms $E^{i,j}$ for which $i < j$ since we can swap the variables if necessary without affecting condition 3.

Although pixels can have multiple labels and in general cannot be represented by binary variables, we can do it for the α-expansion operation. Indeed, any configuration f' within a single α-expansion of the initial configuration f can be encoded by a binary vector $x = \{x_p | p \in \mathcal{P}\}$ where $f'(p) = f(p)$ if $x_p = 0$, and $f'(p) = \alpha$ if $x_p = 1$. Let us denote a configuration defined by a vector x as f^x. Thus, we have the energy of binary variables:

$$\tilde{E}(x) = \tilde{E}_{data}(x) + \tilde{E}_{smoothness}(x) + \tilde{E}_{visibility}(x)$$

where

$$\begin{aligned}
\tilde{E}_{data}(x) &= E_{data}(f^x), \\
\tilde{E}_{smoothness}(x) &= E_{smoothness}(f^x), \\
\tilde{E}_{visibility}(x) &= E_{visibility}(f^x).
\end{aligned}$$

Let's consider each term separately, and show that each satisfies condition (3).

1. Data term.

$$\tilde{E}_{data}(x) = \sum_{\langle p, f^x(p)\rangle, \langle q, f^x(q)\rangle \in I} D(p,q) = \sum_{\langle p,l\rangle, \langle q,l\rangle \in I} T(f^x(p) = f^x(q) = l) \cdot D(p,q)$$

where $T(\cdot)$ is 1 if its argument is true and 0 otherwise. Let's consider a single term $E^{p,q}(x_p, x_q) = T(f^x(p) = f^x(q) = l) \cdot D(p,q)$. Two cases are possible:

1A. $l \neq \alpha$. If at least one of the labels $f(p)$, $f(q)$ is not l then $E^{p,q} \equiv 0$. Suppose that $f(p) = f(q) = l$. Then $E(x_p, x_q)$ is equal to $D(p,q)$, if $x_p = x_q = 0$, and is zero otherwise. We assumed that $D(p,q)$ is non-positive, thus, condition (3) holds:

$$E^{p,q}(0,0) + E^{p,q}(1,1) = D(p,q) \leq 0 = E^{p,q}(0,1) + E^{p,q}(1,0)$$

1B. $l = \alpha$. If at least one of the labels $f(p)$, $f(q)$ is α then $E^{p,q}$ depends only on at most one variable so condition (3) holds. (Suppose, for example, that $f(q) = \alpha$, then $f^x(q) = \alpha$ for any value of x_q and, thus, $E^{p,q}(x_p, 0) = E^{p,q}(x_p, 1)$ and $E^{p,q}(0,0) + E^{p,q}(1,1) = E^{p,q}(0,1) + E^{p,q}(1,0)$.)

Now suppose that both labels $f(p)$, $f(q)$ are not α. Then $E(x_p, x_q)$ is equal to $D(p,q)$, if $x_p = x_q = 1$, and is zero otherwise. We assumed that $D(p,q)$ is non-positive, thus, condition (3) holds:

$$E^{p,q}(0,0) + E^{p,q}(1,1) = D(p,q) \leq 0 = E^{p,q}(0,1) + E^{p,q}(1,0)$$

2. Smoothness term.

$$\tilde{E}_{smoothness}(x) = \sum_{\{p,q\}\in\mathcal{N}} V_{\{p,q\}}(f^x(p), f^x(q)).$$

Let's consider a single term $E^{p,q}(x_p, x_q) = V_{\{p,q\}}(f^x(p), f^x(q))$. We assumed that $V_{\{p,q\}}$ is a metric; thus, $V_{\{p,q\}}(\alpha,\alpha) = 0$ and $V_{\{p,q\}}(f(p), f(q)) \leq V_{\{p,q\}}(f(p), \alpha) + V_{\{p,q\}}(\alpha, f(q))$, or $E^{p,q}(1,1) = 0$ and $E^{p,q}(0,0) \leq E^{p,q}(0,1) + E^{p,q}(1,0)$. Therefore, condition (3) holds.

3. Visibility term.

$$\tilde{E}_{visibility}(x) = \sum_{\langle p, f^x(p)\rangle, \langle q, f^x(q)\rangle \in I_{vis}} \infty$$

$$= \sum_{\langle p, l_p\rangle, \langle q, l_q\rangle \in I_{vis}} T(f^x(p) = l_p \wedge f^x(q) = l_q) \cdot \infty.$$

Let's consider a single term $E^{p,q}(x_p, x_q) = T(f^x(p) = l_p \wedge f^x(q) = l_q) \cdot \infty$. $E^{p,q}(0,0)$ must be zero since it corresponds to the visibility cost of the initial configuration and we assumed that the initial configuration satisfies the visibility constraint. Also $E^{p,q}(1,1)$ is zero (if $x_p = x_q = 1$, then $f^x(p) = f^x(q) = \alpha$ and, thus, the conditions $f^x(p) = l_p$ and $f^x(q) = l_q$ cannot both be true since I_{vis} includes only pairs of 3D-points at different depths). Therefore, condition (3) holds since $E^{p,q}(0,1)$ and $E^{p,q}(1,0)$ are non-negative.

5 Experimental results

Some details of our implementation are below. The planes for labels were chosen to be orthogonal to the main axis of the middle camera, and to increase with depth. The labels are distributed so that disparities depend on labels approximately linearly. The depth difference between planes was chosen in such a way that a label change by one results in disparity change by at most one both in x and y directions.

We selected the labels α 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). We started with an initial solution in which all pixels are assigned to the label having the largest depth.

For our data term D we made use of the method of Birchfield and Tomasi [4] 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). The constant K for the data term was chosen to be 30.

We used a simple Potts model for the smoothness term:

$$V_{\{p,q\}}(l_p, l_q) = \begin{cases} U_{\{p,q\}} & \text{if } l_p \neq l_q \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The choice of $U_{\{p,q\}}$ 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. For pixels $\{p, q\} \in \mathcal{N}$ the term $U_{\{p,q\}}$ was implemented as an empirically selected decreasing function of $\Delta I(p, q)$ as follows:

$$U_{\{p,q\}} = \begin{cases} 3\lambda & \text{if } \Delta I(p,q) < 5 \\ \lambda & \text{otherwise} \end{cases} \tag{5}$$

where $\Delta I(p, q)$ is the average of values $|Intensity(p) - Intensity(q)|$ for all three bands (R,G,B). Thus, the energy depends only on one parameter λ. For different images we picked λ empirically.

We performed experiments on three datasets: the 'head and lamp' image from Tsukuba University, the flower garden sequence and the Dayton sequence. We used 5 images for the Tsukuba dataset (center, left, right, top, bottom), 8 images for the flower garden sequence and 5 images for the Dayton sequence. For the Tsukuba dataset we performed two experiments: in the first one we used four pairs of interacting cameras (center-left, center-right, center-top, center-bottom), and in the second one we used all 10 possible pairs. For the flower garden and Dayton sequences we used 7 and 4 interacting pairs, respectively (only adjacent cameras interact). The table below show image dimensions and running times obtained on 450MHz UltraSPARC II processor. We used the max flow algorithm of [5], which is specifically designed for the kinds of graphs that arise in vision. Source code for this algorithm is available on the web from `http://www.cs.cornell.edu/People/vnk`.

dataset	number of images	number of interactions	image size	number of labels	running time
Tsukuba	5	4	384 x 288	16	369 secs
Tsukuba	5	10	384 x 288	16	837 secs
Flower garden	8	7	352 x 240	16	693 secs
Dayton	5	4	384 x 256	16	702 secs

We have computed the error statistics for the Tsukuba dataset, which are shown in the table below. We determined the percentage of the pixels where the algorithm did not compute the correct disparity (the "Errors" column), or a disparity within ± 1 of the correct disparity ("Gross errors"). For comparison, we have included the results from the best known algorithm for stereo reported in [27], which is the method of [7].

The images are shown in 2, along with the areas where we differ from ground truth (black is no difference, gray is a difference of ± 1, and white is a larger difference). Inspecting the image shows that we in general achieve greater accuracy at discontinuities; for example, the camera in the background and the lamp are more accurate. The major weakness of our output is in the top right corner, which is an area of low texture. The behavior of our method in the presence of low texture needs further investigation.

	Errors	Gross errors
4 interactions	6.13%	2.75%
10 interactions	4.53%	2.30%
Boykov-Veksler-Zabih [7]	9.76%	3.99%

We have also experimented with the parameter sensitivity of our method for the Tsukuba dataset using 10 interactions. Since there is only one parameter, namely λ in equation 5, 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	2	5	10	20	40
Error	4.67%	4.53%	5.28%	5.78%	8.27%
Gross errors	2.43%	2.30%	2.30%	2.23%	2.86%

6 Extensions

It would be interesting to extend our approach to more general camera geometries, for example the case when cameras look at the object in the center from all directions. Labels could then correspond to spheres. More precisely, there would be two labels per sphere corresponding to the closest and farthest intersection points.

6.1 Acknowledgements

This research was supported by NSF grants IIS-9900115 and CCR-0113371, and by a grant from Microsoft Research. We also thank Rick Szeliski for providing us with some of the imagery used in section 5.

Appendix: Edges for terms $E^{i,j}$

In this section we show how to add edges for a term $E^{i,j}$ in the equation 2 assuming that condition (3) holds. This is a special case of a more general construction given in [15].

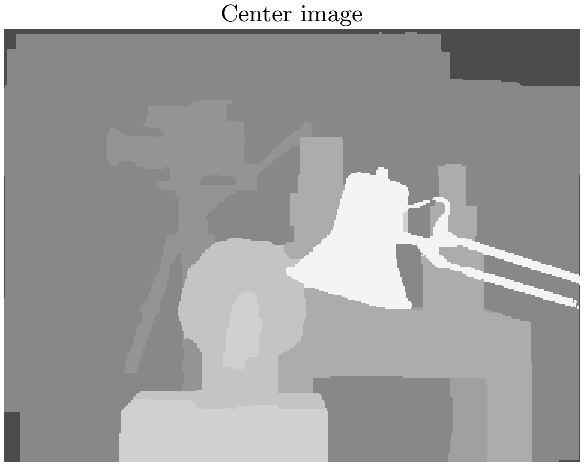
- if $E(1,0) > E(0,0)$ then we add an edge (s, v_i) with the weight $E(1,0) - E(0,0)$, otherwise we add an edge (v_i, t) with the weight $E(0,0) - E(1,0)$;
- if $E(1,0) > E(1,1)$ then we add an edge (v_j, t) with the weight $E(1,0) - E(1,1)$, otherwise we add an edge (s, v_j) with the weight $E(1,1) - E(1,0)$;
- the last edge that we add is (v_i, v_j) with the weight $E(0,1) + E(1,0) - E(0,0) - E(1,1)$.

We have omitted indices i, j in $E^{i,j}$ for simplicity of notation.

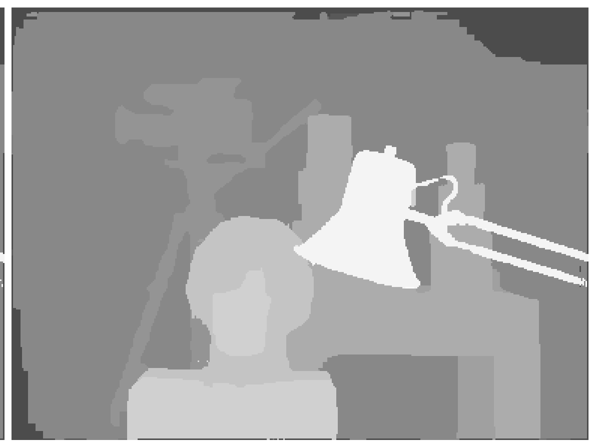
Of course it is not necessary to add edges with zero weights. Also, when several edges are added from one node to another, it is possible to replace them with one edge with the weight equal to the sum of weights of individiual edges.

Center image

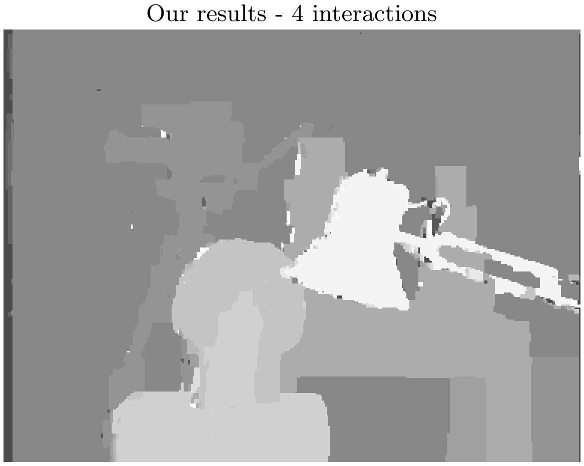
Ground truth



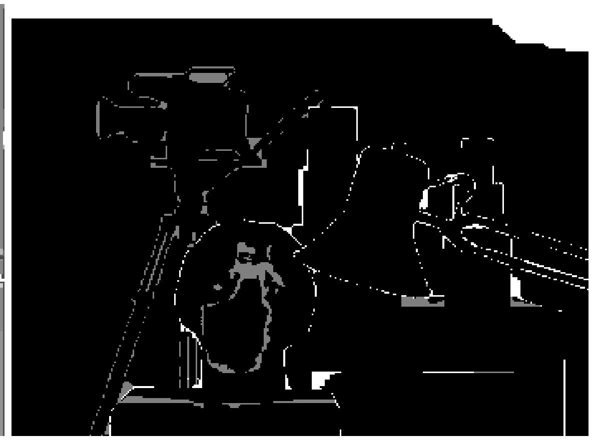
Our results - 4 interactions



Our results - 10 interactions

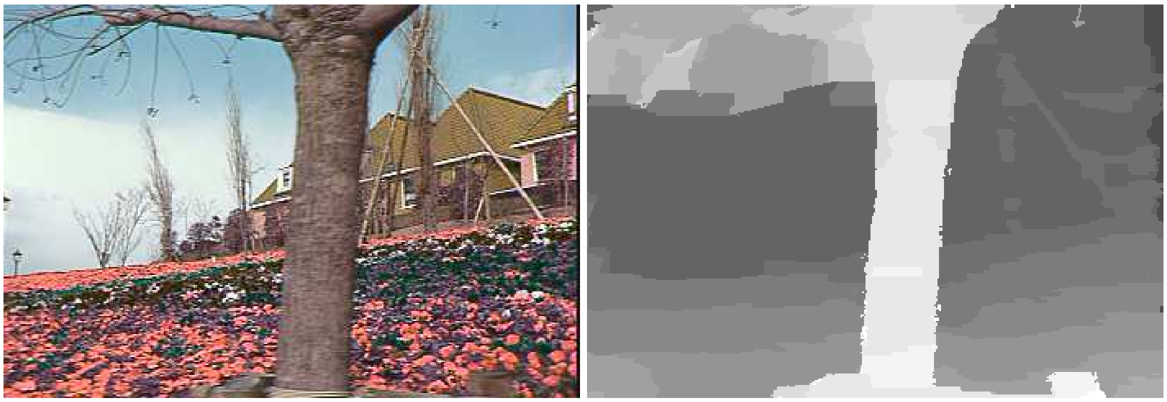


Boykov-Veksler-Zabih results [7]



Comparison of our results with ground truth

Fig. 2. Results on Tsukuba dataset.



Middle image Our results

Fig. 3. Results on the flower garden sequence.

Middle image Our results

Fig. 4. Results on the Dayton sequence.

Appendix: Minimizing our energy function is NP-hard

It is shown in [7] 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 $\mathcal{L}$ and a non-negative function $C : \mathcal{S} \times \mathcal{L} \mapsto \Re^+$. We seek the labeling $f : \mathcal{S} \mapsto \mathcal{L}$ that minimizes

$$E_P(f) = \sum_{p \in \mathcal{P}} C(p, f(p)) + \sum_{\{p,q\} \in \mathcal{N}} T(f(p) \neq f(q)). \tag{6}$$

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 1. 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 C. We will create a new instance of our energy minimization problem as follows. There will be $|\mathcal{L}| + 1$ images. The first one will be the original image $\mathcal{S}$, and for each each label $l \in \mathcal{L}$ there will be an image $\mathcal{P}_l$. For each pixel $p \in \mathcal{S}$ there will be a pixel in $\mathcal{P}_l$ which we will denote as p_l; thus, $|\mathcal{P}_l| = |\mathcal{S}|$ for all labels $l \in \mathcal{L}$.

The set of labels $\mathcal{L}$, the neighborhood system $\mathcal{N}$ and the smoothness function $V_{\{p,q\}}$ will be the same as in the Potts energy minimization problem:

$$V_{\{p,q\}}(l_p, l_q) = T(l_p \neq l_q), \qquad \{p, q\} \in \mathcal{N}$$

The set of interactions I will include all pairs $\{\langle p, l\rangle, \langle p_l, l\rangle\}$, where $p \in \mathcal{S}$, $l \in \mathcal{L}$. The corresponding cost will be

$$D(p, p_l) = C(p, l) - K_p$$

where K_p is a constant which ensures that all costs $D(p, p_l)$ are non-positive (for example, $K_p = \max_{l \in \mathcal{L}} C(p, l)$). There will be no visibility term (i.e. the set of interactions I_{vis} will be empty). The instance of our minimization problem is now completely specified.

Any labeling $f : \mathcal{S} \mapsto \mathcal{L}$ can be extended to the labeling $\bar{f} : \mathcal{P} \mapsto \mathcal{L}$ if we set $\bar{f}(p_l) = l$, $p_l \in \mathcal{P}_l$. It is easy to check that

$$E(\bar{f}) = E_P(f) - K$$

where $K = \sum_{p \in \mathcal{S}} K_p$ is a constant. Moreover, there is no labeling $\bar{f}' : \mathcal{P} \mapsto \mathcal{L}$ with the smaller value of the energy E such that $\bar{f}'(p) = \bar{f}(p)$ for all pixels $p \in \mathcal{S}$. Thus, the global minimum $\bar{f}$ of the energy E will yield the global minimum of the Potts energy E_P (we need to take the restriction of $\bar{f}$ on $\mathcal{S}$).

References

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

2. Stephen Barnard. Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3(1):17–32, 1989.
3. S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *International Conference on Computer Vision*, pages 489–495, 1999.
4. 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.
5. Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 2134 of *LNCS*, pages 359–374. Springer-Verlag, September 2001.
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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.
8. R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision*, 9(2):83–112, November 1992.
9. O.D. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *European Conference on Computer Vision*, 1998.
10. L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
11. 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.
12. H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *European Conference on Computer Vision*, pages 232–248, 1998.
13. S.B. 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.
14. Vladimir Kolmogorov and Ramin Zabih. Visual correspondence with occlusions using graph cuts. In *International Conference on Computer Vision*, pages 508–515, 2001.
15. Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? In *European Conference on Computer Vision*, 2002. Also available as Cornell CS technical report CUCS-TR2001-1857.
16. K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):197–216, July 2000.
17. A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, February 1994.
18. W.N. Martin and J.K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, March 1983.
19. Tomaso Poggio, Vincent Torre, and Christof Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.
20. S. Roy. Stereo without epipolar lines: A maximum flow formulation. *International Journal of Computer Vision*, 1(2):1–15, 1999.
21. S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *International Conference on Computer Vision*, 1998.

22. Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Technical Report 81, Microsoft Research, 2001. To appear in *IJCV*. An earlier version appears in CVPR 2001 Workshop on Stereo Vision.
23. S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):1–23, November 1999.
24. Dan Snow, Paul Viola, and Ramin Zabih. Exact voxel occupancy with graph cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 345–352, 2000.
25. R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics and Image Processing*, 58(1):23–32, July 1993.
26. R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *International Conference on Computer Vision*, pages 517–523, 1998.
27. Richard Szeliski and Ramin 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, September 1999. Springer-Verlag.