Generalized Multi-Camera Scene Reconstruction Using Graph Cuts

Generalized Multi-camera Scene Reconstruction Using Graph Cuts

Vladimir Kolmogorov[1], Ramin Zabih[1], and Steven Gortler[2]

[1] Computer Science Department, Cornell University, Ithaca, NY 14853
[2] Computer Science Department, Harvard University, Cambridge, MA 02138

Abstract. Reconstructing a 3-D scene from more than one camera is a classical problem in computer vision. One of the major sources of difficulty is the fact that not all scene elements are visible from all cameras. In the last few years, two promising approaches have been developed [11, 12] that formulate the scene reconstruction problem in terms of energy minimization, and minimize the energy using graph cuts. These energy minimization approaches treat the input images symmetrically, handle visibility constraints correctly, and allow spatial smoothness to be enforced. However, these algorithm propose different problem formulations, and handle a limited class of smoothness terms. One algorithm [11] uses a problem formulation that is restricted to two-camera stereo, and imposes smoothness between a pair of cameras. The other algorithm [12] can handle an arbitrary number of cameras, but imposes smoothness only with respect to a single camera. In this paper we give a more general energy minimization formulation for the problem, which allows a larger class of spatial smoothness constraints. We show that our formulation includes both of the previous approaches as special cases, as well as permitting new energy functions. Experimental results on real data with ground truth are also included.

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. The problem is quite difficult, in large part because not all scene elements are visible from all cameras.

In this paper, we approach the scene reconstruction problem from the point of view of energy minimization. We build upon two recent algorithms [11, 12] that give an energy minimization formulation of the scene reconstruction problem, and then minimize the energy using graph cuts. Both of these algorithms treat the input images symmetrically, handle visibility constraints correctly, and allow spatial smoothness to be enforced. Moreover, due to the use of graph cuts to perform the energy minimization, they are fast enough to be practical. However, the algorithms [11, 12] use different problem formulations, and handle limited classes of smoothness terms. We propose a new problem energy minimization

approach that includes both these methods as special cases, as well as permitting a larger class of energy functions.

We begin with a review of related work, including a discussion of the algorithms of [11, 12]. In section 4 we give a precise definition of the problem that we wish to solve, and define the energy that we will minimize. Section 5 shows that our problem formulation contains the two previous methods [11, 12] as special cases. In section 6 we describe how to use graph cuts to compute a strong local minimum of our energy. Experimental data is presented in section 7.

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 [16, 21] 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 [6]. 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 [15].

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 [19] and space carving [14].

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 4, 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

It is well known that stereo, like many problems in early vision, can be elegantly stated in terms of energy minimization. The energy minimization problem has traditionally been solved via simulated annealing [2, 8], which is extremely slow in practice.

While energy minimization has been widely used for stereo, only a few papers [10, 17, 20] 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.

In the last few years powerful energy minimization algorithms have been developed based on graph cuts [4, 5, 9, 11, 17]. These methods are fast enough to be practical, and yield quite promising experimental results for stereo [18, 22]. 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 [13] that give graph constructions for a quite general class of energy functions.

Although [17] and [20] use energy minimization via graph cuts, their focus is quite different from ours. [17] 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. [20] 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 also related to the work of [10], which also relies on graph cuts. They extend the work of [5], 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.

3 Graph cut algorithms for scene reconstruction

The results we will present generalize two recent papers: an algorithm for two-camera stereo with occlusions [11], and an algorithm for multi-camera scene reconstruction [12]. Both of these algorithms treat the input images symmetrically, handle visibility constraints correctly, and allow spatial smoothness to be enforced. The major difference between them lies in their problem formulations, and in the class of smoothness terms they permit.

The stereo with occlusions algorithm of [11] uses a problem formulation that is restricted to two cameras. In their representation, a pair of pixels from the two images that may potentially correspond is called an *assignment*. An assignment is *active* when the corresponding scene element is visible in both images. The goal of the algorithm is to find the set of active assignments. There is a hard constraint that a given pixel is involved in at most one active assignment. A pixel that is involved in no active assignments is occluded, and there is a term in the energy function that introduces a penalty for each occluded pixel. Spatial smoothness is imposed with a term that involves assignments; hence, smoothness involves a pair of cameras at a time.

The multi-camera scene reconstruction algorithm given in [12] can handle an arbitrary number of cameras. The problem is represented with a set of depth labels (typically planes). Each pixel in each camera must be assigned a depth label, such that the energy is minimized. Spatial smoothness is imposed with a term that involves a single camera at once.

3.1 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 [7]. 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.

3.2 The expansion move algorithm

Energy minimization algorithms that rely on graph cuts essentially perform a problem reduction. The algorithms with the best performance [5, 11, 12] rely on the expansion move algorithm introduced by [5]. For a given disparity α, an expansion move increases the set of pixels that are assigned the disparity α. The algorithm selects (in a fixed order or at random) a disparity α, and then finds the configuration within a single α-expansion move. If this decreases the energy, then we go there; if there is no α that decreases the energy, we are done. The expansion move algorithm is thus a simple local improvement algorithm, that computes a local minimum in a strong sense: the output is a configuration such that no expansion move can decrease the energy.

The only difficult part of the expansion move algorithm is to find the configuration within a single expansion move that most decreases the energy. This

is done by computing the minimum cut on an appropriately defined graph. The precise details vary depending upon the technical definition of a configuration and the exact form of the energy function.

4 Problem formulation

Now we will formalize the problem we are trying to solve. We will introduce two mappings describing the geometry of the scene, and enforce hard constraints between them. These mappings will be similar to the ones used in [12] and [11], respectively.

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. Our first mapping f will describe depths for all pixels. More formally, the labeling f is a mapping from $\mathcal{P}$ to $\mathcal{L}$ 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*.

Our method has the same limitation as the earlier graph cut multi-camera algorithm [12] and voxel coloring [19]. Namely, 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 $\mathcal{D}(P) < \mathcal{D}(Q)$. 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. Several possible criteria for "closeness" are discussed in [12]. 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 $l_1 = l_2$.

To simplify the notation, we will denote interactions in I as $\langle p, q, l \rangle$ where p, q are pixels and l is a depth label.

Since 3D-points $\langle p, l \rangle$ and $\langle q, l \rangle$ are close to each other, the interaction $\langle p, q, l \rangle$ approximately corresponds to a single point in 3D-space (it can be, for example, the middle point between $\langle p, l \rangle$ and $\langle q, l \rangle$). We can describe the geometry of the scene by specifying which interactions are visible. Let us introduce another mapping $g : I \to \{0, 1\}$. $g(\langle p, q, l \rangle)$ will be 1 if this interaction is visible in both pixels p and q, and 0 otherwise. This mapping allows us to introduce the data term (i.e. the photoconsistency constraint) very naturally: we will enforce

photoconsistency between p and q only if the interaction $\langle p, q, l \rangle$ is *active*, i.e. $g(\langle p, q, l \rangle) = 1$.

The mapping g is very similar to the mapping used in the stereo with occlusions method [11]. Note that in their work each assignment is characterized by a disparity $q_x - p_x$, which generalizes to a depth label in our framework.

4.1 Our energy function

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

$$E(f, g) = E_{data}(g) + E^{(1)}_{smooth}(f) + E^{(2)}_{smooth}(g) + E_{vis}(f) + E_{consistency}(f, g)$$

The terms $E^{(1)}_{smooth}$ and E_{vis} were used in [12]. The terms $E^{(2)}_{smooth}$ and E_{data} are similar to the ones used in [11]. The new term $E_{consistency}$ will enforce consistency between the mappings f and g.

Data term The data term will be

$$E_{data}(g) = \sum_{i \in I} D_i(g(i))$$

where $D_i(0) = K$ for some constant K and $D_i(1)$ depends on intensities of pixels p and q involved in the interaction i. We can have, for example, $D_{\langle p,q,l \rangle}(1) = (Intensity(p) - Intensity(q))^2$.

Smoothness terms Two smoothness terms enforce smoothness on two fields f and g, respectively. They involve a notion of neighborhood; we assume that there are two neighborhood systems: one on pixels

$$\mathcal{N}_1 \subset \{\{p, p'\} \mid p, p' \in \mathcal{P}\}$$

and one on interactions

$$\mathcal{N}_2 \subset \{\{i, i'\} \mid i, i' \in I\}.$$

$\mathcal{N}_1$ can be the usual 4-neighborhood system: pixels p and p' are neighbors if they are in the same image and $|p'_x - p_x| + |p'_y - p_y| = 1$. $\mathcal{N}_2$ can be defined similarly; interactions $\langle p, q, l \rangle$ and $\langle p', q', l \rangle$ are neighbors if p and p' are neighbors (or they are the same pixel): $|p'_x - p_x| + |p'_y - p_y| \leq 1$. The only requirement on $\mathcal{N}_2$ is that neighboring interactions must have the same depth label.

We will write the first smoothness term as

$$E^{(1)}_{smooth}(f) = \sum_{\{p,p'\} \in \mathcal{N}_1} V_{\{p,p'\}}(f(p), f(p'))$$

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 $V(l_1, l_2) = \min(|l_1 - l_2|, R)$ for constant R. Note that this smoothness term involves only a single camera, as does [12].

The second smoothness term can be written as

$$E^{(2)}_{smooth}(g) = \sum_{\{i,i'\} \in \mathcal{N}_2} V_{\{i,i'\}} \cdot T(g(i) \neq g(i'))$$

where $T(\cdot)$ is 1 if its argument is true and 0 otherwise. Note that this smoothness term involves pairs of cameras, as does [11].

Visibility term This 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 for this purpose. Thus, the set I_{vis} can 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 $l' > l$.

Consistency term The last term will enforce consistency between two mappings f and g. It can be formulated as follows: if an interaction $\langle p, q, l\rangle$ is active, then the label for pixels p and q must be l. We can write this as

$$E_{consistency}(f, g) = \sum_{\langle p,q,l\rangle \in I} \infty \cdot T(g(\langle p, q, l\rangle) = 1 \ \wedge \ (f(p) \neq l \ \vee \ f(q) \neq l))$$

5 Relation to previous methods

In this section we show that multi-camera reconstruction algorithm of [12] and the stereo with occlusions algorithm of [11] are special cases of our general framework.

5.1 Multi-camera reconstruction algorithm

Let us consider our energy function with the second smoothness term $E^{(2)}_{smooth}$ omitted. We now show that this is equivalent to the energy used in the multi-camera reconstruction algorithm [12].

We can view our energy as function of only one mapping f if we assume that g is determined from the minimality condition:

$$\tilde{E}(f) = E(f, g(f)) \quad \text{with} \quad g(f) = \arg\min_{g} E(f, g)$$

Let us consider an interaction $i = \langle p, q, l \rangle \in I$. Since g is not involved in the smoothness constraint, the value $g(i)$ will depend only on $f(p)$ and $f(q)$. Namely, if $f(p) \neq l$ or $f(q) \neq l$ then $g(i)$ must be 0 because of the consistency constraint between f and g. Now suppose that $f(p) = f(q) = l$. In this case the value $g(i)$ will be determined from the minimality condition: it will be 0 if $D_i(0) < D_i(1)$, and it will be 1 if $D_i(0) > D_i(1)$. Thus, the data term for interaction i becomes

$$\tilde{E}_{data(i)}(f) = D_i(0) + D(p, q) \cdot T(f(p) = f(q) = l)$$

where $D(p, q) = min(D_i(1) - D_i(0), 0)$. This is exactly the expression for data term used in [12] except for the constant $D_i(0)$.

5.2 Stereo with occlusions algorithm

Now let us consider our energy with the first smoothness term $E^{(1)}_{smooth}$ omitted. We will show that in the case of stereo our formulation is equivalent to the stereo with occlusions algorithm [11].

As before, we will view our energy as a function of only one mapping g (with f determined from the minimality condition). It is easy to see that the smoothness term $E^{(2)}_{smooth}$ is equivalent to the smoothness term used in [11], and the sum of two terms $E_{vis}(f(g)) + E_{consistency}(f(g), g)$ is equivalent to the hard uniqueness constraint in [11]. (The uniqueness constraint says that each pixel can be involved in at most one active assignment).

[11] has an additional term which basically counts the number of occlusions. However, it is easy to see that having a penalty C for an occlusion is equivalent to having a penalty $C/2$ for an interaction (or assignment) being inactive. Thus, our data term term is equivalent to the sum of data and occlusion terms in [11], which concludes the argument.

6 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, g) and a disparity α. Another configuration (f', g') is defined to be within a single α-*expansion* of (f, g) if two conditions are satisfied:

– All pixels must either keep their depth labels, or change it to α. In other words, for any pixel $p \in \mathcal{P}$ either $f'(p) = f(p)$ or $f'(p) = \alpha$.
– All inactive interactions whose depth is different from α must remain inactive. In other words, for any interaction $\langle p, q, l \rangle \in I$ conditions $g(\langle p, q, l \rangle) = 0$ and $l \neq \alpha$ imply $g'(\langle p, q, l \rangle) = 0$.

This notion of an expansion move was proposed by [5], and forms the basis for several very effective stereo algorithms [5, 10, 11].

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) that gives the largest decrease in the energy $E(f, g)$. 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 [5, 11].

One restriction on the algorithm is that the initial configuration must satisfy the visibility and consistency constraints (i.e. the initial energy must be finite). This will guarantee that all subsequent configurations will have finite energies, i.e. they will satisfy these constraints as well.

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.

6.1 Energy minimization using graph cuts

Instead of doing an explicit problem reduction, we will use a result from [13] which says that for energy functions of binary variables of the form

$$E(x_1, \ldots, x_n) = \sum_i E^i(x_i) + \sum_{i<j} E^{i,j}(x_i, x_j) \tag{1}$$

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

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0). \tag{2}$$

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 $E^i(x_i)$ and $E^{i,j}(x_i, x_j)$ we add edges as described in [13].

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.

6.2 α-expansion

In this section we will show how to convert our energy function into the form of equation 1. 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 2.

In an α-expansion, active interactions may become inactive, and inactive interactions whose depth is α may become active. Suppose that we start off with an initial configuration (f^0, g^0) satisfying the visibility and consistency constraints. The active interactions for a new configuration within one α-expansion will be a subset of $I^0 \cup I^\alpha$, where $I^0 = \{\langle p, q, l\rangle \in I \mid g^0(\langle p, q, l\rangle) = 1 \text{ and } l \neq \alpha\}$ and $I^\alpha = \{\langle p, q, \alpha\rangle \in I\}$.

It is easy to see that any configuration (f, g) within a single α-expansion of the initial configuration (f^0, g^0) can be encoded by two binary vectors $x = \{x_p \mid p \in \mathcal{P}\}$ and $y = \{y_i \mid i \in I^\alpha \cup I^0\}$. We will use the following formula for correspondence between binary vectors and configurations:

$$\begin{aligned}
\forall p \in \mathcal{P} \quad & f(p) = \begin{cases} f^0(p) & \text{if } x_p = 0 \\ \alpha & \text{if } x_p = 1 \end{cases} \\
\forall i \in I^0 \quad & g(i) = 1 - y_i \\
\forall i \in I^\alpha \quad & g(i) = y_i \\
\forall i \notin I^0 \cup I^\alpha \quad & g(i) = 0
\end{aligned}$$

Let us denote a configuration defined by vectors (x, y) as (f^x, g^y). We now have the energy of binary variables:

$$\tilde{E}(x, y) = \tilde{E}_{data}(y) + \tilde{E}^{(1)}_{smooth}(x) + \tilde{E}^{(2)}_{smooth}(y) + \tilde{E}_{vis}(x) + \tilde{E}_{consistency}(x, y)$$

where

$$\begin{aligned}
\tilde{E}_{data}(y) &= E_{data}(g^y), \\
\tilde{E}^{(1)}_{smooth}(x) &= E^{(1)}_{smooth}(f^x), \\
\tilde{E}^{(2)}_{smooth}(y) &= E^{(2)}_{smooth}(g^y), \\
\tilde{E}_{vis}(x) &= E_{vis}(f^x), \\
\tilde{E}_{consistency}(x, y) &= E_{consistency}(f^x, g^y).
\end{aligned}$$

We can now consider each term separately, and show that each satisfies condition (2).

1. Data term.

$$\tilde{E}_{data}(y) = \sum_{i \in I^0} D_i(1 - y_i) + \sum_{i \in I^\alpha} D_i(y_i)$$

Condition (2) is satisfied since each term in this sum depends only on one variable.

2. First smoothness term.

$$\tilde{E}^{(1)}_{smooth}(x) = \sum_{\{p,p'\}\in\mathcal{N}_1} V_{\{p,p'\}}(f^x(p), f^x(p')).$$

Let's consider a single term $E^{p,p'}(x_p, x_{p'}) = V_{\{p,p'\}}(f^x(p), f^x(p'))$. We assumed that $V_{\{p,p'\}}$ is a metric; thus, $V_{\{p,p'\}}(\alpha,\alpha) = 0$ and $V_{\{p,p'\}}(f(p), f(p')) \le V_{\{p,p'\}}(f(p),\alpha) + V_{\{p,p'\}}(\alpha, f(p'))$, or $E^{p,p'}(1,1) = 0$ and $E^{p,p'}(0,0) \le E^{p,p'}(0,1) + E^{p,p'}(1,0)$. Therefore, condition (2) holds.

3. Second smoothness term.

$$\tilde{E}^{(2)}_{smooth}(y) = \sum_{\{i,i'\}\in\mathcal{N}_2} V_{\{i,i'\}} \cdot T(g^y(i) \neq g^y(i'))$$

Let's consider a single term $E^{i,i'}(y_i, y_{i'}) = V_{\{i,i'\}} \cdot T(g^y(i) \neq g^y(i'))$. Since the depths of i and i' are the same, they either both belong to I^0 or both belong to I^α. In both cases condition $g^y(i) \neq g^y(i')$ is equivalent to condition $y_i \neq y_{i'}$. Thus, $E^{i,i'}(0,0) = E^{i,i'}(1,1) = 0$ and $E^{i,i'}(0,1) = E^{i,i'}(1,0) = V_{\{i,i'\}} \ge 0$, so condition (2) holds.

4. Visibility term.

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

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

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 (2) holds since $E^{p,q}(0,1)$ and $E^{p,q}(1,0)$ are non-negative.

5. Consistency term.

$$\tilde{E}_{consistency}(x,y) = \sum_{\langle p,q,l\rangle \in I} \infty \cdot T(g^y(\langle p,q,l\rangle) = 1 \ \wedge \ (f^x(p) \neq l \ \vee \ f^x(q) \neq l))$$

The term involving interaction $i = \langle p,q,l\rangle$ can be rewritten as the sum $E^{p,i}(x_p, y_i) + E^{q,i}(x_q, y_i)$ where $E^{p,i}(x_p, y_i) = \infty \cdot T(g^y(i) = 1 \ \wedge \ f^x(p) \neq l)$ and $E^{q,i}(x_q, y_i) = \infty \cdot T(g^y(i) = 1 \ \wedge \ f^x(q) \neq l)$. Let's consider one of the terms, for example $E^{p,i}$. Two cases are possible:

5A. $l \neq \alpha$. If $f^0(p) \neq l$ then $E^{p,i} \equiv 0$, otherwise $E^{p,i}(x_p, y_i) = \infty \cdot T(y_i = 0 \ \wedge \ x_p = 1)$, so $E^{p,i}(1,0) = \infty$ and $E^{p,i}(0,0) = E^{p,i}(1,1) = E^{p,i}(0,1) = 0$.

5B. $l = \alpha$. In this case $E^{p,i}(x_p, y_i) = \infty \cdot T(y_i = 1 \wedge x_p = 0)$, so $E^{p,i}(0,1) = \infty$ and $E^{p,i}(0,0) = E^{p,i}(1,1) = E^{p,i}(1,0) = 0$.

7 Experimental results

We performed experiments for the two special cases discussed in section 5. We will refer to the case in section 5.1 as "algorithm I" and the case in section 5.2 as "algorithm II".

We used the same datasets used in [12]: the "head and lamp" image from Tsukuba University, the flower garden sequence and the Dayton sequence. We also used the same geometry, i.e. depth labels, interaction sets I and I_{vis} and the neigborhood system $\mathcal{N}_1$ for algorithm I. Our choice of the neighborhood system $\mathcal{N}_2$ for algorithm II is a slight variation of that of [11]: interactions $\langle p_1, q_1, l\rangle$ and $\langle p_2, q_2, l\rangle$ are neighbors if pixels p_1 and p_2 in a specified camera are neighbors according to $\mathcal{N}_1$.

Our choice of parameters for algorithms I and II is the same as in [12] and [11], respectively. In both cases the energy depends only on one parameter λ, which we picked empirically for different datasets. As in [12], we stop after three iterations.

The results for algorithm II on the flower garden and Dayton datasets contain scattered pixels with no depth labels (or, more precisely, assigning any depth label to such pixels results in the same value of the energy function). Such pixels are probably due to the high noise in these datasets (or their miscalibration). We performed some postprocessing of the results: we assign to such pixels the label of the closest labeled pixel.

The table below show dataset sizes, number of interacting pairs of cameras that we used, and running times obtained on 450MHz UltraSPARC II processor. For all datasets we used 16 depth labels. The max flow implementation we used is one specifically designed for the kinds of graphs that arise in vision [3].

dataset	number of images	number of interactions	image size	running time (I)	running time (II)
Tsukuba	5	4	384 x 288	369 secs	532 secs
Tsukuba	5	10	384 x 288	837 secs	1584 secs
Flower garden	8	7	352 x 240	693 secs	680 secs
Dayton	5	4	384 x 256	702 secs	481 secs

We have computed the error statistics for the Tsukuba dataset, which are shown in the table below.

	Errors	Gross errors
4 interactions (I)	6.13%	2.75%
4 interactions (II)	5.02%	1.40%
10 interactions (I)	4.53%	2.30%
10 interactions (II)	5.30%	2.36%
Boykov-Veksler-Zabih [5]	9.76%	3.99%

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 [22], which is the method of [5].

The images are shown in figure 1. The image at bottom right shows the areas where algorithm I differs 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.

8 Conclusions and Future Work

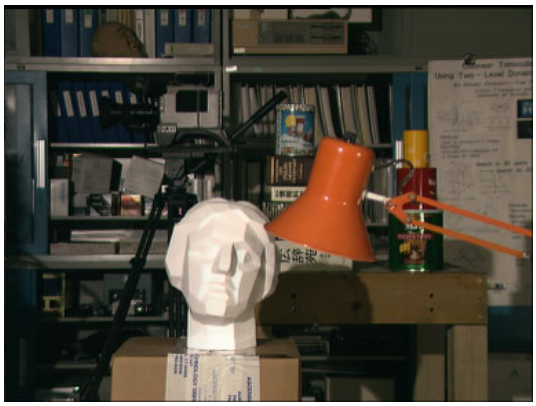
We have described a new energy minimization framework for multi-camera scene reconstruction. The energy can be efficiently minimized using graph cuts, and gives good experimental results. Furthermore, the new framework generalizes two previous algorithms, as well as permitting new energy functions that combine two distinct kinds of spatial smoothness constraints. More work is needed to determine if these new energy functions have experimental advantages over the previous methods that we have generalized.

Acknowledgements

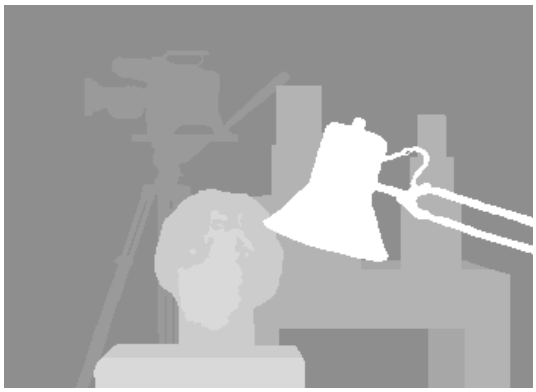
This research was supported by the National Science Foundation under grant IIS-9900115.

References

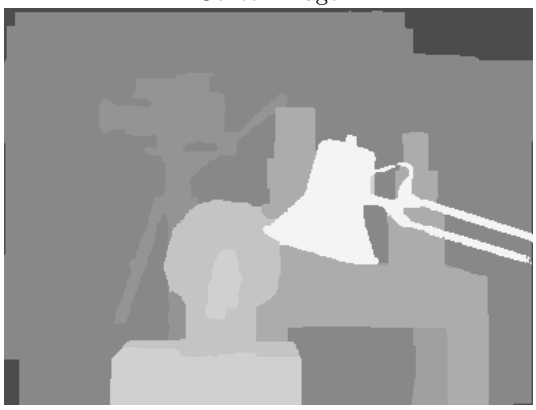
1. Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.
2. Stephen Barnard. Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3(1):17–32, 1989.
3. Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. In *Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 2134 of *LNCS*, pages 359–374, September 2001.
4. 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.
5. 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.
6. R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision*, 9(2):83–112, November 1992.
7. L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
8. 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.



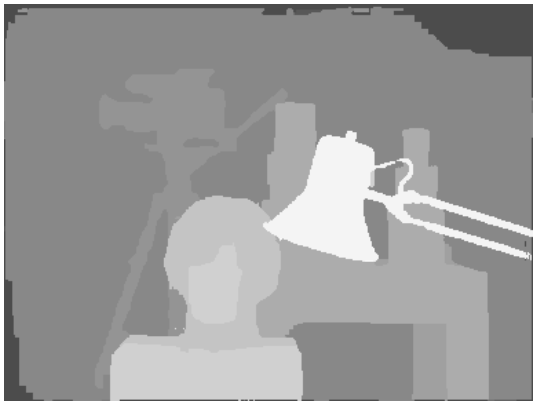
Center image



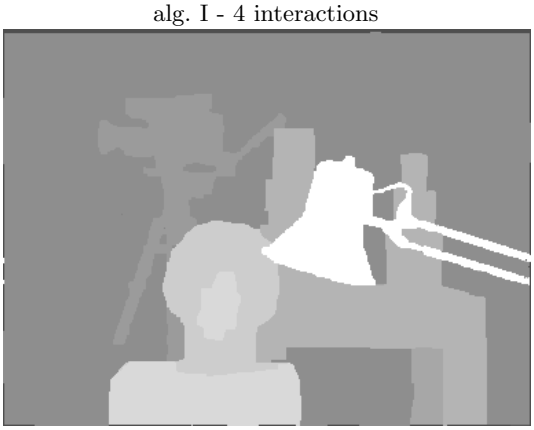
Ground truth



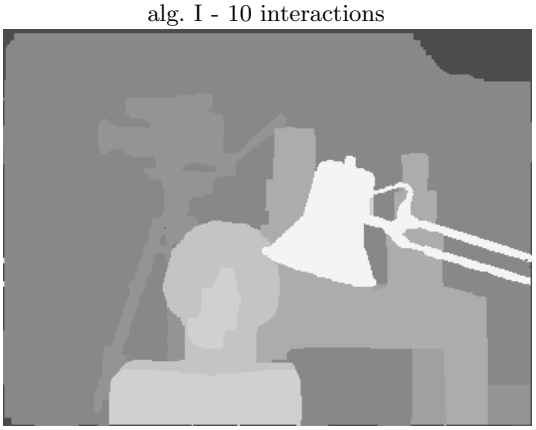
alg. I - 4 interactions



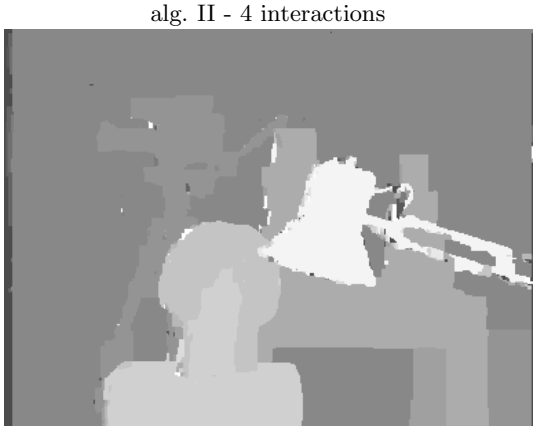
alg. I - 10 interactions



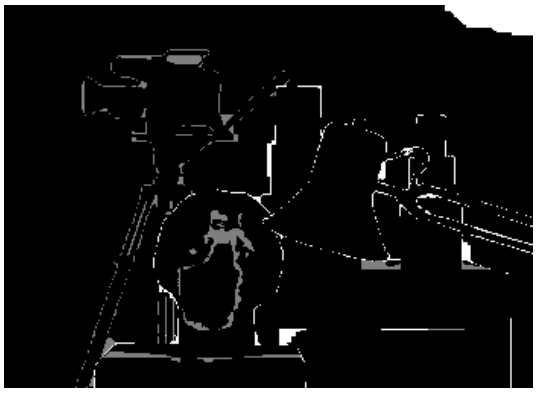
alg. II - 4 interactions



alg. II - 10 interactions

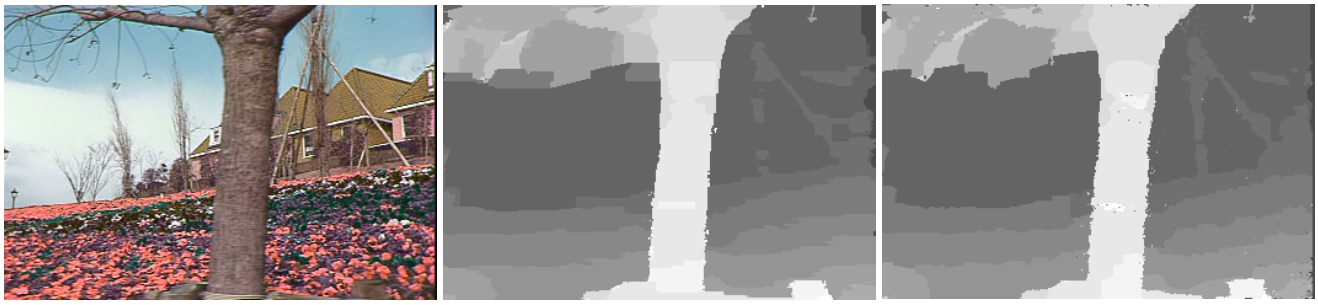


Boykov-Veksler-Zabih results [5]



Comparison of alg. I result with ground truth

Fig. 1. Results on Tsukuba dataset.



Middle image alg. I result alg. II result

Fig. 2. Results on the flower garden sequence.

Middle image alg. I result alg. II result

Fig. 3. Results on the Dayton sequence.

9. H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *European Conference on Computer Vision*, pages 232–248, 1998.
10. 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.
11. Vladimir Kolmogorov and Ramin Zabih. Visual correspondence with occlusions using graph cuts. In *International Conference on Computer Vision*, pages 508–515, 2001.
12. Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision*, volume 3, pages 82–96, 2002.
13. Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? In *European Conference on Computer Vision*, volume 3, pages 65–81, 2002. Revised version to appear in *IEEE Transactions on Pattern Analysis and Machine Intelligence*
14. 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.
15. 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.
16. 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.
17. S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *International Conference on Computer Vision*, 1998.
18. Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, April 2002.
19. S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):1–23, November 1999.
20. 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.
21. R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics and Image Processing*, 58(1):23–32, July 1993.
22. 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.