International Journal of Computer Vision 34(2/3), 147–161 (1999)

Stereo Without Epipolar Lines: A Maximum-Flow Formulation

SÉBASTIEN ROY*
NEC Research Institute, 4 Independence Way, Princeton, NJ 08540, USA
sebastien@research.nj.nec.com

Abstract. This paper describes a new algorithm for solving the stereo correspondence problem with a global 2-d optimization by transforming it into a maximum-flow problem in a graph. This transformation effectively removes explicit use of epipolar geometry, thus allowing direct use of multiple cameras with arbitrary geometries. The maximum-flow, solved both efficiently and globally, yields a minimum-cut that corresponds to a disparity surface for the whole image at once. This global and efficient approach to stereo analysis allows the reconstruction to proceed in an arbitrary volume of space and provides a more accurate and coherent depth map than the traditional stereo algorithms. In particular, smoothness is applied uniformly instead of only along epipolar lines, while the global optimality of the depth surface is guaranteed. Results show improved depth estimation as well as better handling of depth discontinuities. While the worst case running time is $O(n^{1.5}d^{1.5}\log(nd))$, the observed average running time is $O(n^{1.2}\,d^{1.3})$ for an image size of n pixels and depth resolution d.

Keywords: stereo correspondence, maximum-flow, multiple cameras, range estimation, 3-d reconstruction

1. Introduction

It is well known that depth-related displacements in stereo pairs always occur along lines associated with the camera motion, the epipolar lines. These lines reduce the stereo correspondence problem to one dimension and the ordering constraint allows dynamic programming to be applied (Baker, 1981; Ohta and Kanade, 1985; Cox et al., 1996; Faugeras, 1993). However, it is clear that this reduction to 1-d is an oversimplification of the problem, primarily required to enforce smoothness constraints in a computationally efficient way. The solutions obtained on consecutive epipolar lines can vary significantly and create artifacts across epipolar lines, especially affecting object boundaries that are perpendicular to the epipolar lines (e.g. vertical object boundary with horizontal epipolar lines).

In this paper, we address the full 2-d matching problem, eliminating the need for explicit epipolar lines and replacing the traditional ordering constraint with the more general *local coherence* constraint. To perform the global 2-d optimization, we cast the stereo correspondence problem as a maximum-flow problem in a graph and show how the associated minimum-cut can be interpreted as a disparity surface. While the theoretical worst case computational complexity is significantly higher for maximum-flow than dynamic programming, in practice, the average case performance is similar. We also show how this new approach, being based on 2-d optimization, allows both binocular and n-camera stereo configurations, as well as arbitrary 3-d reconstruction volumes.

There have been several earlier attempts to relate the solutions of consecutive epipolar lines matched with dynamic programming. In Ohta and Kanade (1985), dynamic programming is used to first match epipolar lines and then iteratively improve the solutions obtained by using vertical edges as reference. In Cox et al. (1996), a probabilistic approach is used to relate the individual matchings obtained by dynamic programming to improve the depth map quality. First, it proposes to improve a given epipolar line matching by using the previous line solution to improve its own

*Visiting from Université de Montréal, Département d'informatique et de recherche opérationnelle, C.P. 6128, Succ. Centre-Ville, Montréal, Québec, H3C 3J7.

solution. However, this introduces a non-desirable vertical asymmetry. A second approach is to iteratively improve each epipolar line solutions with its neighboring lines solution. While this *local* approach is not globally optimal, it provides an efficient way to introduce smoothness constraints across epipolar lines. In Belhumeur (1996), a Bayesian approach to the stereo correspondence problem is described. The resulting optimization problem can be solved efficiently by using dynamic programming along epipolar lines, resulting in the same problem as (Ohta and Kanade, 1985; Cox et al., 1996) of relating the independent solutions. It proposes a heuristic method called *iterated stochastic dynamic programming* that uses previously computed adjacent epipolar line solutions to iteratively improve randomly selected solutions. This approach is not globally optimal and furthermore introduces a large amount of smoothness that tends to blur depth discontinuities.

The concept of using maximum-flow appeared in Greig et al. (1989) in the context of binary Markov Random Fields, where each pixel of a binary image is given one of two labels. The maximum-flow formulation for more than two labels and a convex discontinuity cost was presented by Roy and Cox (1998) in the context of stereoscopic correspondence. Recently, Ishikawa and Geiger (1998) presented a similar method as Roy and Cox (1998), but expressed in the context of Markov Random Fields and applied to image segmentation. Also, Boykov et al. (1998) presented a Markov Random Field formulation with non-linear discontinuity costs that give rise to a minimum multi-way cut problem. They present an approximate method based on efficient maximum-flow steps applied to binary subproblems.

Some multiple-cameras algorithms have been presented (see Faugeras, 1993; Cox, 1994; Kang et al., 1994; Kanade et al., 1996). In Cox (1994), a pair of camera is used as a *reference* or base pair. Other cameras provide extra information to enrich the matching cost function of the reference camera pair. The matching then proceeds using dynamic programming as in Cox et al. (1996). In Kang et al. (1994) and in Kanade et al. (1996), a multiple-camera real-time stereo system is presented. They use a single *reference* camera to perform the matching. All the other cameras provide the information pertinent to each possible depth of points in the reference image. The depth is computed independently for each pixel, making it impossible to enforce a smoothness constraint between pixels. Instead, the images are low-pass filtered and the matching process uses windows rather than single pixel values. While this achieves some level of smoothness in the solution, it has the undesirable side effect of blurring the depth discontinuities.

Section 2 describes a general stereo framework to be used with multiple images from arbitrary viewpoints and arbitrary reconstruction volumes. It also describes a simple stereo matching cost function that supports those multiple images. In Section 3, the stereo problem is extended from matching single epipolar lines to solving for a full disparity map, making use of the *local coherence* constraint. In Section 4, the stereo matching problem is formulated as a maximum-flow problem. Details of the maximum-flow algorithm and performance issues are presented in Section 4.3. Experiments on both classic two-image and multiple-image stereo sequence are presented and discussed in Section 5.

2. The Stereo Framework

This section describes a general stereo framework. It consists of two distinct parts. First, a volume of the 3-d world is selected to constrain where the stereo matching actually occurs. Any resulting reconstructed surface must lie inside that volume. Second, each 3-d world point inside the matching volume is projected onto the set of images to provide pixel intensity values. This information is then used to derive the matching cost necessary to perform stereo analysis. Even though it is performed inside a 3-d volume of space, our algorithm always recovers a depth surface that cuts this volume in two parts, and not an arbitrary 3-d shape inside the volume.

2.1. The Stereo Matching Space

The volume of 3-d space that contains every possible depth surface is referred to as the *matching space* and has been used before in stereo (see Yang and Yuille (1995) and Marr and Poggio (1979)). This volume is discretized and searched by the stereo algorithm for an optimal depth surface. It is characterized by *front* and *back* regions that must be disjoint. By definition, a valid stereo depth surface always separates the *front* and *back* of the matching space, and is therefore defined as a function of the *front* (or *back*).

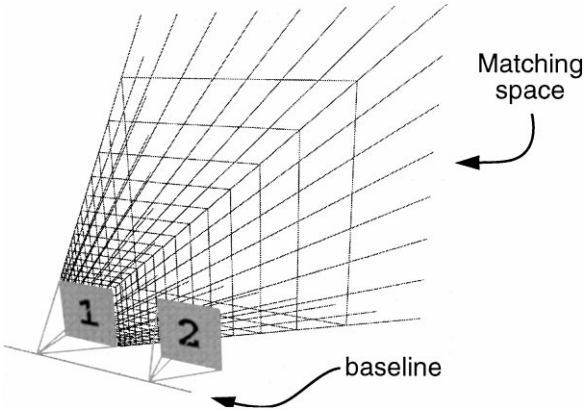

Figure 1. Standard stereo framework. Two horizontally separated cameras with parallel optical axes. The stereo matching volume is the viewing volume of camera 1.

For standard stereo, the matching space is a truncated pyramid corresponding to the viewing volume of a camera (as in Fig. 1). The front and back are simply the near and far planes of the viewing pyramid. Obviously, any valid surface (separating the front and near planes) will yield exactly one disparity value for every pixel of the selected camera.

In order to be solved using this stereo algorithm, there is no other restriction placed on the matching space other than to possess a front and a back. This implies that arbitrary chunks of the world can be analyzed and the recovered surfaces can be fully or partially closed, depending on the dimensionality and relationship of the front and back regions. For the purpose of this paper, we selected a partition of space that only allows open surfaces with uniform quantization of either disparity or depth, as depicted in Fig. 2.

The matching space is defined as a projective 3-d volume (to allow pyramids as well as cubes) formed

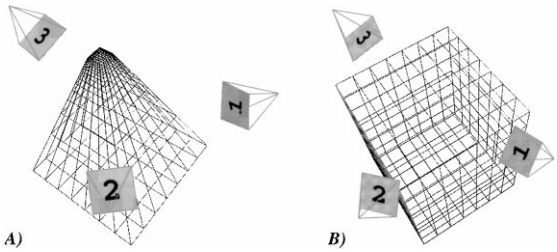

Figure 2. General stereo framework. Three cameras at arbitrary positions and orientations in 3-d space, around two types of matching spaces, (A) with uniform disparity steps and (B) with uniform depth steps.

by three axes a, b, and d containing respectively a'_{size}, b'_{size}, and d'_{size} quantized steps, that is

$$\begin{bmatrix} a' \\ b' \\ d' \\ 1 \end{bmatrix} \quad \text{with} \quad \begin{array}{lll} a' & \in \mathbb{N}, & 0 \le a' < a'_{size} \\ b' & \in \mathbb{N}, & 0 \le b' < b'_{size} \\ d' & \in \mathbb{N}, & 0 \le d' < d'_{size} \end{array}$$

where a' and b' intuitively correspond to a pixel coordinate inside a viewing volume such as in Fig. 1 while d' corresponds to the disparity or depth of that pixel.

A point (a', b', d') is expressed in the 3-d world as an homogeneous point $\mathbf{p}_w$ defined as

$$\mathbf{p}_w = \mathbf{Q} \begin{bmatrix} a' \\ b' \\ d' \\ 1 \end{bmatrix} \tag{1}$$

where $\mathbf{Q}$ is a 4×4 matrix that allows for changing the shape and position of the matching space in the world.

In particular, the matching space is made identical to the viewing volume of a camera (see Fig. 2A) by defining $\mathbf{Q}$ as

$$\mathbf{Q} = \mathbf{W}^{-1} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{bmatrix} \times \begin{bmatrix} \frac{x_{size}}{a'_{size}-1} & & & 0 \\ & \frac{y_{size}}{b'_{size}-1} & & 0 \\ & & \frac{d_{\max}-d_{\min}}{d'_{size}-1} & d_{\min} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where x_{size} and y_{size} represent the image size, $d_{\min}$ and $d_{\max}$ are the allowed disparity interval, and where $\mathbf{W}$ is the 4×4 viewing transformation matrix of the camera. Notice that d' is moved to the fourth row, making it represent disparity rather than depth, as would be the case for standard stereo with uniformly quantized disparities.

Similarly, if a uniform quantization of depth is desired (see Fig. 2B), the last row of $\mathbf{Q}$ should be

[0, 0, 0, 1], as in this definition

$$\mathbf{Q} = \begin{bmatrix} \frac{a_{\max}-a_{\min}}{a'_{size}-1} & & & a_{\min} \\ & \frac{b_{\max}-b_{\min}}{b'_{size}-1} & & b_{\min} \\ & & \frac{d_{\max}-d_{\min}}{d'_{size}-1} & d_{\min} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the intervals $[a_{\min}, a_{\max}]$, $[b_{\min}, b_{\max}]$, and $[d_{\min}, d_{\max}]$ represent the span of the matching space position in the world. Notice that in this case, the matching space is defined independently of the camera geometries.

2.2. Pixel Intensity Values

In this section, we present a general framework to handle stereo in the context of multiple images taken under arbitrary camera geometries. It naturally extends the traditional two-image, single-baseline framework for stereo. In this context, each camera i must simply provide a single matrix W_i transforming a point in the world coordinate system into a point in the camera image space. This matrix is usually obtained from carefully measured camera parameters, which are classified as *internal* and *external*. Internal parameters are linked to the optics of the camera, such as focal length, aspect ratio, lense distortion, etc... External parameters are the position and orientation of the camera in the world. For the purpose of stereo, detailled knowledge of all the parameters, or *strong calibration*, is not necessary. By finding only a few corresponding pair of points, we can directly compute the *fundamental matrices* (Faugeras, 1993) that express the relationship between the cameras, yeilding the transformation matrix W_i without resorting to a full calibration. In contrast to *strong calibration*, the disparities obtained by stereo matching with these *weakly calibrated* cameras do not have a known relation to the real depth in the scene.

A set of n *inspection* cameras $C_1, \ldots, C_n$ provides n images $I_1, \ldots, I_n$ of a scene, as depicted in Fig. 3 (with $n = 3$). A *cube* (not shown in Fig. 3) provides the matching volume where we wish to compute the depth surface. Inside the matching volume, a cube point (a', b', d') can be transformed to the homogeneous image point $\mathbf{p}_i$ in the image of camera i by the relation

$$\begin{aligned} \mathbf{p}_i &= \mathbf{J}\,\mathbf{W}_i\,\mathbf{p}_w \\ &= \mathbf{J}\,\mathbf{W}_i\,\mathbf{Q}\,[a' \quad b' \quad d' \quad 1] \end{aligned}$$

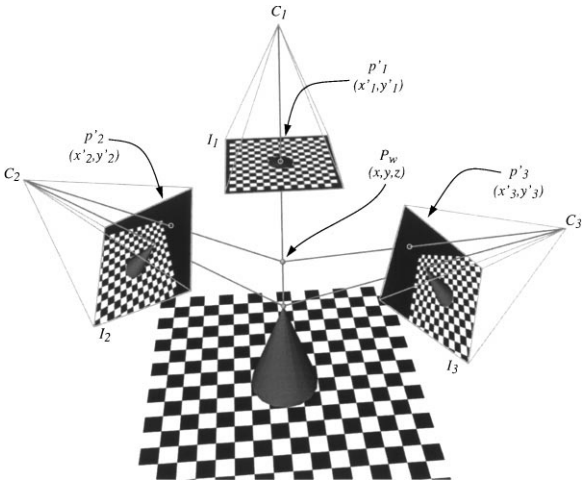

Figure 3. Multiple-camera stereo setup. You can back-project any world point $\mathbf{p}_w$ to each inspection camera (C_1, C_2, C_3), obtaining the set of image points (p'_1, p'_2, p'_3).

where $\mathbf{W}_i$ is a 4×4 matrix describing the camera geometry, $\mathbf{Q}$ is from Eq. (1), and $\mathbf{J}$ is a simple 3×4 projection matrix

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

From a transformed and projected point $\mathbf{p}_i$, the corresponding image coordinates $\mathbf{p}'_i$ are obtained from the relation

$$\mathbf{p}'_i = H(\mathbf{p}_i)$$

where H is a homogenizing function

$$H\left(\begin{bmatrix} x \\ y \\ h \end{bmatrix}\right) = \begin{bmatrix} x/h \\ y/h \end{bmatrix}.$$

The pixel intensity vector $\mathbf{v}_{(a',b',d')}$ associated to each cube point (a', b', d') is defined as

$$\begin{aligned} \mathbf{v}_{(a',b',d')} = \{I_i(H(\mathbf{J}\,\mathbf{W}_i\,\mathbf{Q}\,[a' \quad b' \quad d' \quad 1]^T)), \\ \forall i \in [1, \ldots, n]\} \end{aligned} \tag{2}$$

where $I_i([x' \; y']^T)$ is the intensity of pixel $[x' \; y']^T$ in image i. This vector contains all the pixel intensity information from the inspection cameras for a particular value of (a', b', d').

2.3. Matching Cost

In order to perform stereo matching, a *matching cost* function is required. Ideally, it is minimum for a likely match and large for an unlikely one. Deriving a matching cost that represents well the stereo problem is not a trivial task. Deriving one that can also be globally minimized in polynomial time is even more difficult. Until now, dynamic programming provided an efficient way to minimize cost functions that enforce smoothness, which are generally viewed as very appropriate for the stereo problem. However, as a side effect of this method, the cost function had to be *weakened* by enforcing smoothness along a line instead of a surface. In this paper, the maximum-flow minimization method removes this limitation while remaining efficient and therefore can solve better suited cost functions than previously possible. There is however a new restriction on the cost function: the smoothness term must be convex, rather than arbitrary for the dynamic programming approach. This, as experiments will show, is not a major problem and does not significantly *weaken* the cost function. This new cost function is described next.

If we assume that surfaces are lambertian (i.e. their intensity is independent of the viewing direction) then the pixel intensity values, components of $\mathbf{v}_{(a',b',d')}$, should be identical when (a', b', d') is on the surface of an object and thus a valid match. Then, we can naturally define the matching cost $cost(a', b', d')$ as the L_2-norm of the pixel intensity vector $\mathbf{v}_{(a',b',d')}$, that is

$$cost(a', b', d') = \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{v}_{(a',b',d')_i} - \overline{\mathbf{v}_{(a',b',d')}} \right)^2. \quad (3)$$

where $\overline{\mathbf{v}_{(a',b',d')}}$ is the mean of the components of $\mathbf{v}_{(a',b',d')}$.

3. Recovering a Full Disparity Map

Typically, stereo matching is performed independently along epipolar lines, to allow an efficient algorithm to be used, such as dynamic programming. A natural extension to matching a single pair of epipolar lines at a time would be to extend it to the whole image at once, as depicted in Fig. 4, by essentially matching all pairs of epipolar lines simultaneously. The matching volume is quantized in three dimensions with two axes (a, b) representing image pixels and an axis d for the disparity associated with each pixel (a, b). The depth surface contains all the computed disparities of the base image. The goal of this construction is to take advantage

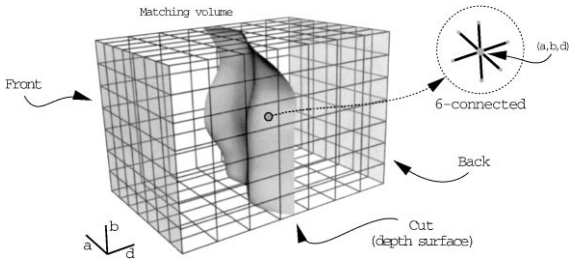

Figure 4. Matching whole images. In the matching volume, the `Front` and `Back` correspond respectively to the minimum and maximum disparities. The depth surface cuts the matching volume in two parts, isolating `Front` and `Back`.

of one very important property of disparity fields, *local coherence*, which suggests that disparities tend to be locally very similar in all directions, including *across* epipolar lines.

Dynamic programming cannot be used anymore to globally establish correspondence since there is no two-dimensional ordering that can be used in a way similar to the use of the one-dimensional ordering along individual epipolar lines.

Many solutions for global disparity surface matching have been proposed (Cox, 1994; Ohta and Kanade, 1985; Belhumeur, 1996). Typically, these algorithms propose an approach in which a solution is iteratively improved by using the previous matching obtained for neighboring epipolar lines. While this can sometimes work in practice, these solutions are not very efficient and not optimal with regard to their inability to find the global minimum of the cost function they are minimizing.

4. Stereo Matching as a Maximum Flow Problem

We propose to solve globally for the disparity surface by adding a source and a sink to the formulation of Fig. 4, and treat it as a flow problem in a graph, as depicted in Fig. 5. The graph depicts a flow network where each arc has a stated flow capacity, and each node acts as a junction. The flow entering a node is always equal to the flow leaving a node, thereby enforcing a *flow conservation* property. The maximum-flow problem we wish to solve is concerned about finding the largest flow that can leave the source and reach the sink through the graph, without exceeding the capacities of the arcs (Cormen et al., 1990). According to the *Max-flow min-cut theorem* (Cormen et al., 1990), the set of edges that are saturated by the maximum flow through the graph represents a *minimum-cut* of

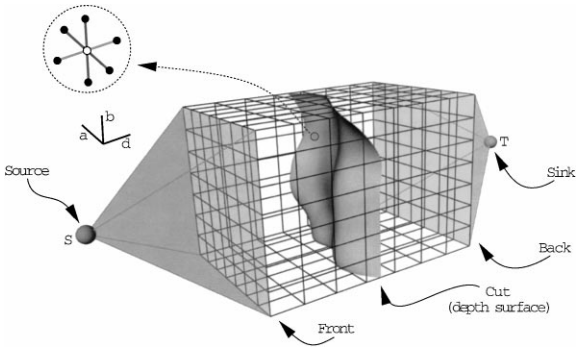

Figure 5. Image Matching as a Maximum Flow problem.

the graph. By connecting the source and sink respectively to the *front* and *back* of the matching volume, as in Fig. 5, a cut separating the source and sink effectively represents a disparity surface. Moreover, a minimum-cut will represent the minimum cost disparity surface sought.

Consider the graph $G = (V, E)$ forming a 3-d mesh as in Fig. 5. The vertex set V is defined as

$$V = V^* \cup \{s, t\}$$

where s is the source, t is the sink, and V^* is the 3-d mesh

$$V^* = \{(a', b', d') : 0 \le a' < a'_{size},\ 0 \le b' < b'_{size},\ 0 \le d' < d'_{size} + 1\}$$

where (a'_{size}, b'_{size}) is the base image size and d'_{size} is the depth or disparity solution space. It is made larger by one node to provide a dummy node required for an appropriate graph formulation.

Internally the mesh is six-connected. There are two disjoint sets of vertices, V_{front} and V_{back}, that represent the *front* and *back* of the graph, such that the source s is connected to each node of V_{front}, while each node of V_{back} is connected to the sink t. We define them as

$$V_{front} = \{(a', b', 0) : 0 \le a' < a'_{size}, 0 \le b' < b'_{size}\}$$
$$V_{back} = \{(a', b', d'_{size}) : 0 \le a' < a'_{size}, 0 \le b' < b'_{size}\}.$$

The edges of the graph are defined as

$$E = E_{label} \cup E_{penalty} \cup E_{in} \cup E_{out}$$

with

$$\begin{aligned} E_{in} &= \{(s, u) : u \in V_{front}\} \\ E_{out} &= \{(u, t) : u \in V_{back}\} \\ E_{label} &= \{(u, v) \in V^* \times V^* : u - v = (0, 0, \pm 1)\} \\ E_{penalty} &= \big\{(u, v) \in V^* \times V^* : u_{d'} = v_{d'} \\ &\qquad \text{and } (v_{a'}, v_{b'}) \in \mathcal{N}_{(u_{a'}, u_{b'})}\big\} \end{aligned}$$

where $(u_{a'}, u_{b'}, u_{d'})$ and $(v_{a'}, v_{b'}, v_{d'})$ are the a', b' and d' components of nodes u and v, and $\mathcal{N}_{(a',b')}$ is the neighborhood of pixel (a', b'). It can be arbitrarily chosen, but in this paper we use the 4-neighborhood

$$\mathcal{N}_{(a',b')} = \{(a' \pm 1, b'), (a', b' \pm 1)\}$$

With respect to Fig. 5, the edge set E_{in} is the section connecting the source and the front, while E_{out} is the section connecting the back and the sink. The set E_{label} expresses the pixel matching costs and contains all edges parallel to the d axis. The set $E_{penalty}$ expresses the smoothness constraint and contains all edges inside the (a, b) planes. After the minimum cut is obtained, cut edges belonging to $E_{penalty}$ will be discarded while those from E_{label} will represent the obtained disparity. The different role of *label* and *penalty* edges will be further described in Sections 4.1 and 4.2.

We define the edge capacities in the graph in a straightforward way. The connections to the source or the sink have infinite capacities. Each vertex (a', b', d') in the graph corresponds to a potential match that assigns disparity d' to pixel (a', b'), so we can use Eq. (3) to derive its matching cost. This cost is directly used as the capacity of the label edge ($\in E_{label}$) associated to this vertex. To express smoothness, a constant capacity is given to penalty edges ($\in E_{penalty}$). The edge capacity $c(u, v)$ from node u to v is thus defined as

$$c(u, v) = \begin{cases} 0 & \text{if } (u, v) \notin E \\ \infty & \text{if } (u, v) \in E_{in} \text{ or } (u, v) \in E_{out} \\ cost(u) & \text{if } (u, v) \in E_{label} \text{ and } u_{d'} < v_{d'} \\ cost(v) & \text{if } (u, v) \in E_{label} \text{ and } u_{d'} > v_{d'} \\ K & \text{if } (u, v) \in E_{penalty} \end{cases} \tag{4}$$

where K is a smoothness factor.

The minimum-cut C_{min} obtained by computing the maximum-flow over G contains a set of edges of minimum total capacity that isolates the source and the sink.

Being a minimum-cut, $C_{\min}$ is defined as

$$\arg\min_{C}\left(\sum_{(u,v)\in C} c(u,v)\right)$$
$$= \arg\min_{C}\left(\sum_{(u,v)\in C_{label}} c(u,v) + \sum_{(u,v)\in C_{penalty}} c(u,v)\right)$$

where $C_{label} = C \cap E_{label}$ and $C_{penalty} = C \cap E_{penalty}$. We define the labelling functions $L_{(a',b')}$ as the smallest d' component of the nodes of an edge in C_{label}. Such an edge is of the form

$$\big((a', b', L_{(a',b')}), (a', b', L_{(a',b')} + 1)\big)$$

or

$$\big((a', b', L_{(a',b')} + 1), (a', b', L_{(a',b')})\big).$$

Notice that these two forms have the same capacity: $cost(a', b', L_{(a',b')})$. By replacing the edge capacities according to Eq. (4) we have

$$\min_{C}\left(\sum_{(u,v)\in C_{label}} cost(u) + \sum_{(u,v)\in C_{penalty}} K\right)$$
$$= \min_{C}\left(\sum_{((a',b',L_{(a',b')}),v)\in C_{label}} cost\big(a', b', L_{(a',b')}\big) + \sum_{((a',b',L_{(a',b')}),v)\in C_{penalty}} K\right)$$
$$= \sum_{\forall(a',b')} cost\big(a', b', L^*_{(a',b')}\big) + \frac{K}{2}\sum_{\forall(a',b'),\forall(i',j')\in\mathcal{N}_{(a',b')}} \big|L^*_{(a',b')} - L^*_{(i',j')}\big| \qquad (5)$$

where $L^*_{(a',b')}$ is the labelling function associated to the minimum-cut C_{min}. This transformation is possible since a minimum-cut has the property that for all (a', b') there exists exactly one disparity $L_{(a',b')}$ such that the edge $((a', b', L_{(a',b')}), (a', b', L_{(a',b')} + 1))$ belongs to C_{label}. This property is discussed in Section 4.2.

We can see that the number of penalty edges in the minimum cut is directly related to the difference in disparity between neighboring pixels. This is intuitively explained by noticing that when two neighboring pixels have different disparities, a hole is created in the cut between the two adjacent label edges. It must be *patched* by adding to the cut a number of penalty edges equal to the size of the gap, which is the difference in disparities between the pixels.

In summary, the cost function of Eq. (5) corresponds to finding a disparity surface that globally minimizes a *pixel matching cost* term and a *smoothness* term that assigns a linear penalty to a jump in disparity between neighboring pixels. The tradeoff between these terms is determined by the factor K.

4.1. *Expressing Smoothness Through Edge Capacity*

From the partition of E in two sets of edges, the set of penalty edges $E_{penalty}$ is used to control the level of smoothness of the disparity surface (second term of Eq. (5). As depicted in Fig. 6, penalty edges consists of all edges not oriented along the disparity axis d. As shown in Eq. (4), the matching cost defines the capacity of a label edge, while penalty edges are given the constant value K which also corresponds intuitively to an *occlusion* cost. In Fig. 6, the darker edges connecting the black vertices are penalty edges and the lighter edges are label edges. A higher occlusion cost (i.e. larger K) increases the smoothness of recovered surfaces while, inversely, a lower occlusion cost facilitates depth discontinuities.

The effect of the smoothness parameter K is illustrated by a 2-d example problem with a simple cost function, as shown in Fig. 7. The minimum-cut of this simple graph is computed for different smoothness values (0, 1, and ∞) and displayed in Fig. 7 as thick edges. Notice that the label edges, which determine the solution, are horizontal. These extreme values of the smoothness parameter K have intuitive consequences. Setting $K = 0$, each row of the graph is independently given a disparity, therefore achieving maximal discontinuity in the disparity surface. When $K = \infty$, the resulting disparity surface is flat (maximally smooth) and features a single disparity value for the whole image. For $K = 1$, a balance is reached between the matching cost and the smoothness required.

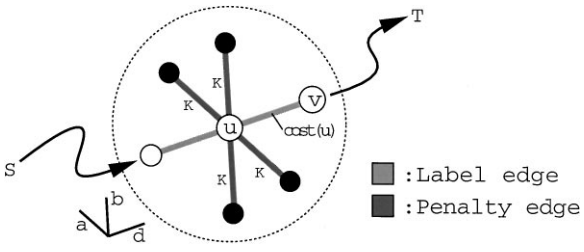

Figure 6. Expressing smoothness through edge capacity.

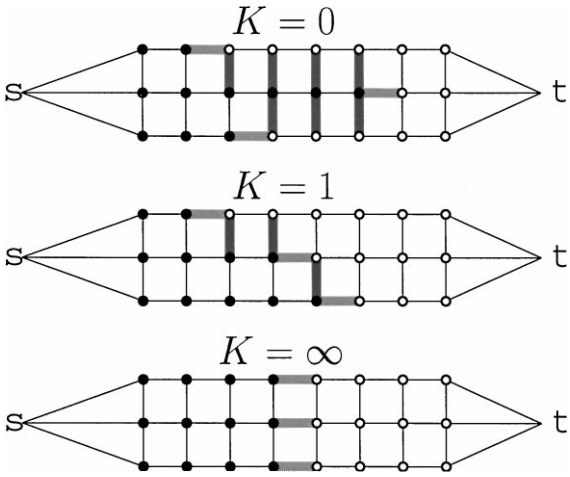

Figure 7. Example cuts for different smoothness values. $K = 0$, maximal discontinuity. $K = 1$, intermediate smoothness. $K = \infty$, infinite smoothness.

4.2. *From a Cut to a Disparity Surface*

The max-*flow* min-*cut* theorem states that once the maximum flow is found, a minimum-cut $C_{\min}$ separates the source and the sink in such a way that the sum of edge capacities of $C_{\min}$ is minimized. This cut is therefore the globally optimal way to separate the source and the sink for our particular cost function. To derive our cost function of Eq. (5), we used an important property of our graph formulation, namely that the minimum cut is guaranteed to provide exactly one depth estimate for each image point, or more simply that the cut does not *fold* on itself. This property can be guaranteed in various ways.

First, we conjecture that a capacity function with a constant capacity K for all penalty edges, such as in Eq. (4), always satisfies this property.

As proposed by Boykov et al. (1998), a large constant can be added to the likelihoods of Eq. (3). The capacity function (Eq. (4)) becomes

$$c(u, v) = \begin{cases} \cdots & \\ cost(u) + B & \text{if } (u, v) \in E_{label} \text{ and } u_{d'} < v_{d'} \\ cost(v) + B & \text{if } (u, v) \in E_{label} \text{ and } u_{d'} > v_{d'} \end{cases}$$

where B is a value larger than the sum of all the penalty edges of the graph. This will effectively add a constant to the energy function of Eq. (5), thereby insuring that the minimum of the modified energy function is the same as the original one.

Also, Ishikawa and Geiger (1998) suggested to assign an infinite capacity to label edges returning from the sink toward the source. The capacity function is modified from Eq. (4) to become

$$c(u, v) = \begin{cases} \cdots & \\ cost(u) & \text{if } (u, v) \in E_{label} \text{ and } u_{d'} < v_{d'} \\ B & \text{if } (u, v) \in E_{label} \text{ and } u_{d'} > v_{d'} \end{cases}$$

where B is infinity (Ishikawa and Geiger, 1998). In fact, it is sufficient and more practical to define B as a value larger than the sum of all the penalty edges of the graph.

Notice that the two previous solutions (Boykov et al., 1998; Ishikawa and Geiger, 1998) do not make any assumption about the capacities of penalty edges. This adds flexibility to the choice of these capacities and might prove to be useful in the future.

The full disparity surface can now be constructed easily from the minimum cut C of graph G as follows. For each point (a', b'), the disparity is $L^*_{(a',b')}$ since the edge $(a', b', L^*_{(a',b')}) - (a', b', L^*_{(a',b')} + 1)$ belongs to $C_{\min}$, as stated in Eq. (5).

4.3. *Solving the Maximum Flow Problem*

There is an abundant literature on algorithms to solve the maximum-flow problem (Cormen et al., 1990; Goldberg and Rao, 1997). For this paper, we implemented a well known algorithm, *preflow-push lift-to-front* (Cormen et al., 1990). Currently, the best maximum-flow algorithm is presented by Goldberg and Rao (1997) and is particularly well suited for sparse graphs like the ones built for stereo matching.

The number of vertices v in the graph is equal to the number of image pixels multiplied by the depth resolution. For an image of total size $s = ab$ pixels, i.e. of dimension $a \times b$, and a depth resolution of d steps, we have $v = sd$. Since the graph is a three-dimensional mesh where each vertex is six-connected,[1] the number of edges e is $e \approx 6sd$.

This implies that the preflow-push algorithm used, with a running time

$$O(ve \log(v^2/e))$$

yields for our problem

$$O(s^2 d^2 \log(sd)).$$

The algorithm with the currently best bound (Goldberg and Rao, 1997) runs in

$$O\left(e^{\frac{3}{2}} \log(v^2/e)\right) \log(U))$$

where U is the largest edge capacity, in our case a constant since pixels have finite values, yielding a running time of

$$O(s^{1.5}d^{1.5} \log(sd) \log(U)).$$

However, we did not use this algorithm in practice since this performance improvement is for the worst case only, and not for the average case. No significant improvement in the average case is expected over the preflow-push relabel algorithm we used.

The dynamic programming approach on separate epipolar lines proposed by Cox et al. (1996) requires a total running time of $\Theta(sd)$, which might seem much better than the maximum-flow algorithm. However, the topology of the graph, the positions of the source and sink, and the structure of edge capacities all tend to make the problem easier to solve, making the average running time much better than the worst case analysis would suggest. Figure 8 shows the typical performance as a function of total image size s (in pixels) and depth resolution d. The average running time is $O(s^{1.2} d^{1.3})$, which is almost linear with respect to image size s (in pixels) and compares favorably with the dynamic programming approach. The typical running time for 256×256 images is anywhere between 1 and 30 min, on a 160 MHz Pentium computer, depending on the depth resolution used. While this is considerably slower than Cox et al. (1996), which was originally built for speed, our algorithm was not optimized for speed. Performance improvement is expected in the future.

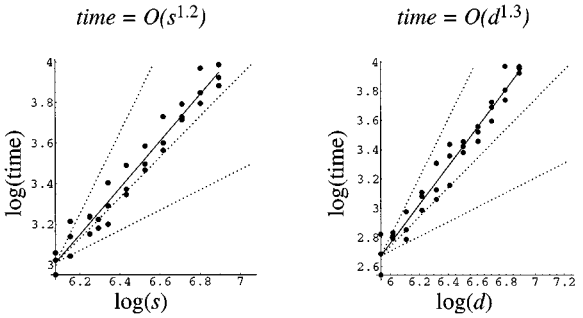

Figure 8. (left) Performance as a function of image size s in pixels, for fixed depth resolution. (right) Performance as a function of depth resolution d for a fixed size s. Three dotted lines show performance levels of $O(\sqrt{s})$, $O(s)$, and $O(s^2)$.

5. Experiments and Results

In this section, results of binocular and n-camera stereoscopic matching from maximum-flow are presented and compared with two other algorithms, both based on dynamic programming. The requirement to support multiple images is not readily handled by the vast majority of stereo algorithms, making many comparisons unpractical.

First, the algorithm referred to as standard stereo uses line-by-line dynamic programming on n-camera with variable depth resolutions. It differs from the maximum-flow algorithm only in the way it computes the disparity surface. They are otherwise identical and their results use the same disparity scale and are not equalized. By equalization, we refer to a solution-dependent transformation, usually non-linear, applied to the solution in order to improve the contrast of the displayed results. The most common such equalization is *histogram equalization*. Often, this transformation makes fair comparison of results very difficult, if at all possible.

Second, the algorithm referred to as MLMH+V is the efficient dynamic programming implementation from Cox et al. (1996) (for the binocular version) and from Cox (1994) (for the n-camera version). It differs from the previous algorithm in that performs an iterative optimization of its disparity solution to enforce smoothness across disparity lines. It should be noted that the results from this algorithm use a different disparity scale (gray levels) than maximum-flow or standard stereo and are equalized to improve their contrast.

Random Dot Stereogram. To demonstrate the symmetry in the disparity map achieved by maximum-flow, we applied it on a random-dot stereogram (see Fig. 9) with disparities set at 0, 4 and 8 pixels. The resulting disparity maps, shown in Fig. 10, differ mostly around depth discontinuities. Maximum-flow features similar boundaries in all directions while standard stereo yields very different boundary shapes, due to the fact that solutions are computed horizontally and no information is shared vertically.

Granite. Figure 11 presents the camera and scene setup for a synthetic sequence of 5 views of a smooth

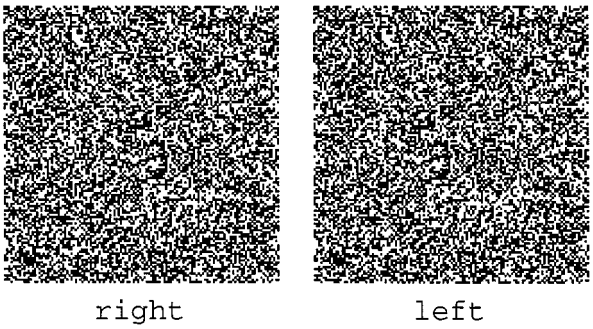

Figure 9. A random dot stereogram (displayed for cross-eyed stereo viewing).

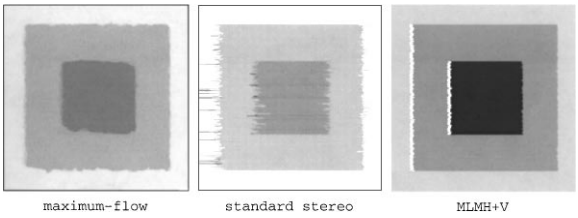

Figure 10. Disparity map for random dot stereogram.

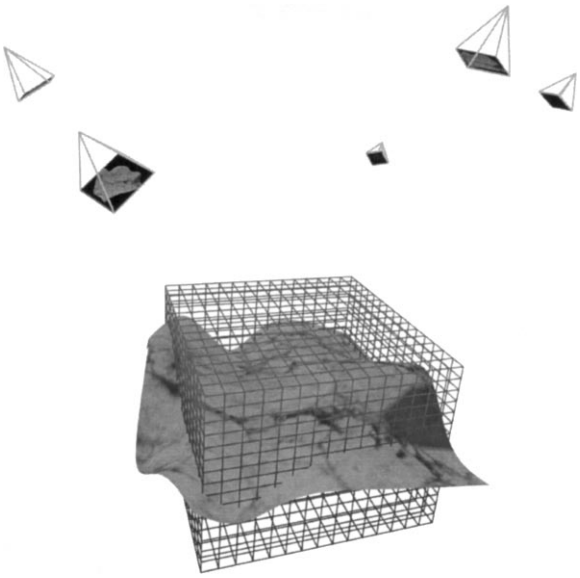

Figure 11. The Granite scene and camera setup. The mesh represents the matching volume.

textured surface. The camera images, displayed in Fig. 12, are put in correspondence over the matching space shown as the 3-d mesh of Fig. 11.

Results for different number of images and different smoothness values are shown in Figs. 13 and 14. The case $K = 0$ corresponds to using direct search to solve for depth and yields a noisy depth map. Figure 14 presents the accuracy of the depth map as a function of smoothness value, for 2 and 5 cameras. Not surprisingly, these curves suggest that better depth map accuracy is achieved with more images used for matching. Also, enforcing some degree of smoothness, even a small amount, is always better than none at all ($K = 0$). Finally, the accuracy degrades slowly as the smoothness is increased to large levels. This implies that the maximum-flow method is very tolerant of bad estimation of the smoothness parameter.

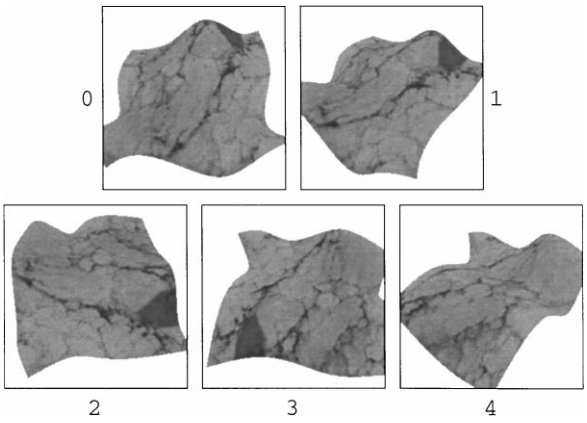

Figure 12. The Granite camera images (256 × 256).

Shrub. Figure 15 shows a pair of the Shrub image sequence (courtesy of T. Kanade and T. Nakahara of CMU). The results in Fig. 16 show how maximum-flow tends to extract sharp and precise depth discontinuities, while standard stereo and MLMH + V produce many artifacts along vertical depth discontinuities. Two levels of depth resolutions are shown (32 and 128 steps) with different level of smoothness. It is notable that even at high smoothness levels, maximum-flow does not produce spurious horizontal links across the gap between the two larger shrubs. The results of multiple-camera analysis are shown in Fig. 17. All the images of this sequence share a common horizontal baseline. Even though the algorithms use different number of images (4 and 7), the total spanned camera displacement is the same and therefore provides about the same depth discrimination. Some image normalization is performed for MLMH + V prior to matching. None was used for the other two algorithms.

Pentagon. The stereo pair Pentagon is shown in Fig. 18. The matching results are presented in Fig. 19.

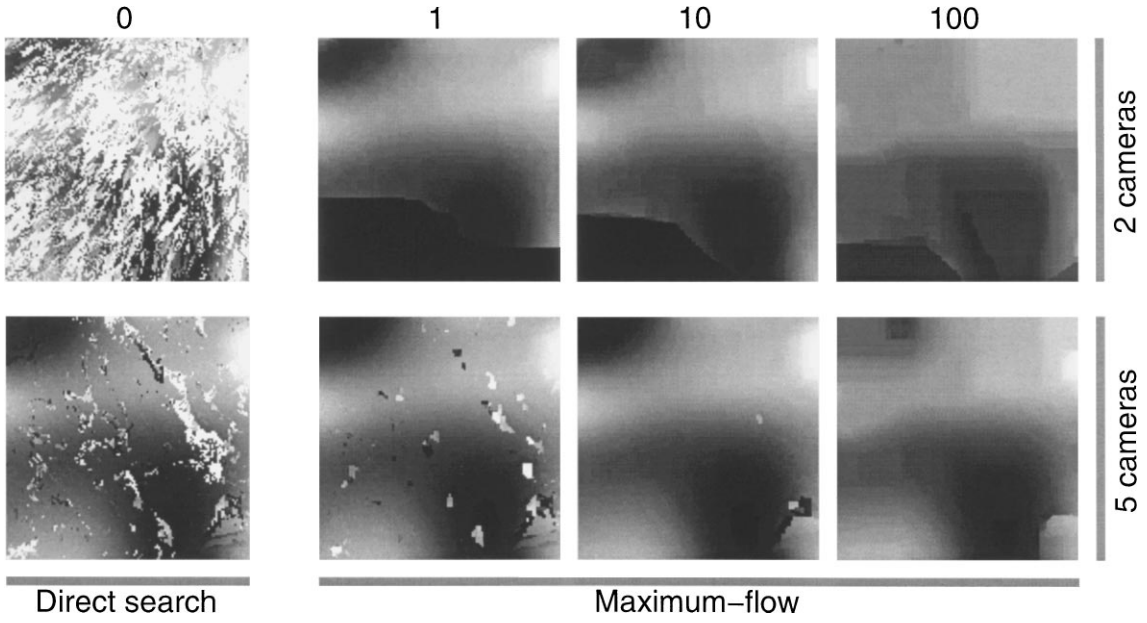

Figure 13. The Granite results. Results shown for smoothness factors 0 to 100, for 2 and 5 cameras.

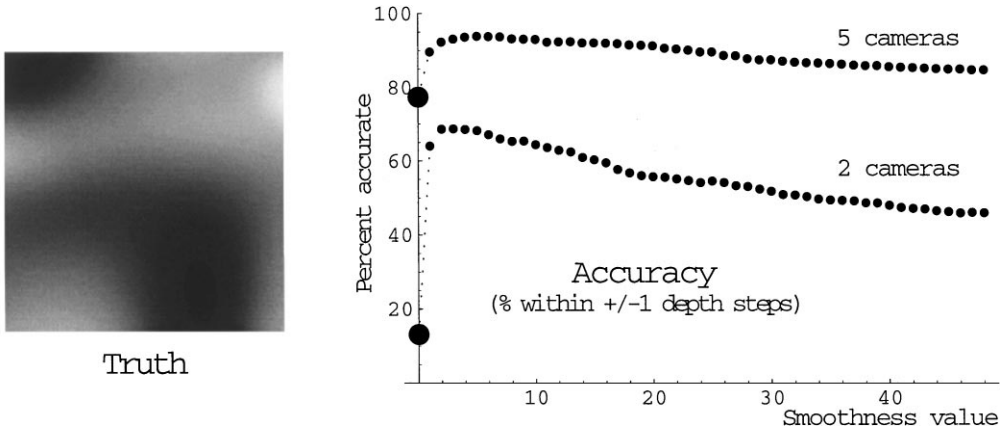

Figure 14. The Granite results.

This stereo pair presents a challenge since the camera motion is not exactly horizontal and contains some rotation, creating image motions that violate the epipolar constraint. Fortunately, algorithms like MLMH+V resist these misalignments better since they allow negative disparities as well as positive. This explains how the highway structures at the top left are well recovered for MLMH+V while the other algorithms produced some noticeable spurious mismatches. As predicted, maximum-flow does produce a more symmetric result, with less spurious horizontal streaks.

Park Meter. The image sequence Park meter shown in Fig. 20 was analyzed for different numbers of images. The results of the binocular case are presented in Fig. 21. Here a number of vertical objects show the difficulties that standard stereo and MLMH+V have to relate horizontal epipolar line solutions. No horizontal streaks are present in the results obtained by maximum-flow. Using 4 images (horizontally displaced along a single baseline), the results shown in Fig. 22 improve significantly from those of Fig. 21. No results were available for MLMH+V.

Figure 15. The Shrub stereo pair.

Roof. The image sequence Roof (courtesy of T. Kanade and E. Kawamura of CMU) is shown in Fig. 23. It contains 13 images featuring either horizontal or vertical translations. The results for maximum-flow and MLMH+V are presented in Fig. 24. The disparity map obtained by maximum-flow is very detailed. In particular, the structure of the roof is well reconstructed. Figure 25 presents a 3-d reconstruction of the Roof sequence based on the maximum-flow depth map. It demonstrates that fine details can be very effectively recovered.

Castle. The sequence Castle from CMU is shown in Fig. 26 and contains 11 images with various combinations of horizontal, vertical and forward camera motion. The 11 images were used to create the disparity map shown on the right for the image shown on the left. A high level of detail and very few spurious matches are present.

It is important to note that this sequence represents a challenge since the actual disparity range, that is, the difference in disparity between the closest and the farthest object, is only 2.7 pixels. Performed at a depth resolution of 96 steps, this implies that the disparity precision achieved is 0.03 pixels.

5.1. Level of Smoothness

In this section, we wish to illustrate how the level of smoothness, represented by the parameter K

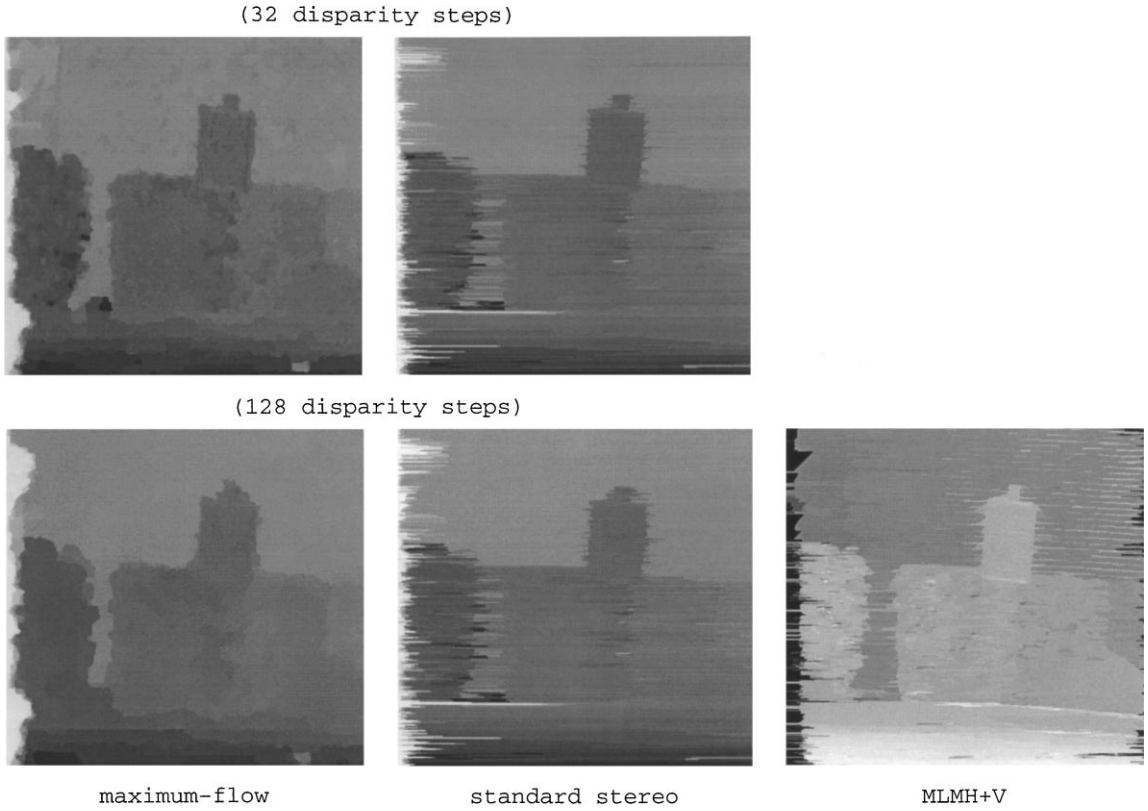

Figure 16. Disparity maps for the Shrub a two precision level (32 and 128 disparity steps). On the left, the maximum-flow results. In the middle and right, results for standard stereo and MLMH+V respectively.

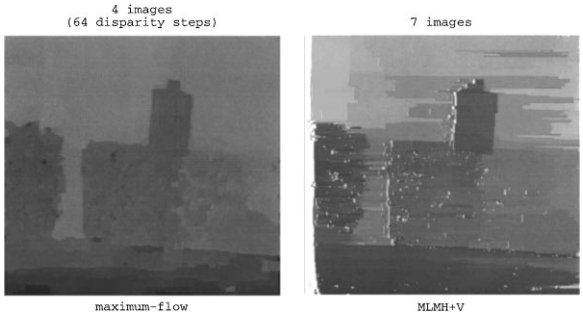

Figure 17. Disparity maps for the Shrub sequence using 4 and 7 images. Both sequences span the same total horizontal displacement and should yield similar results. White points on the right denote detected occlusions.

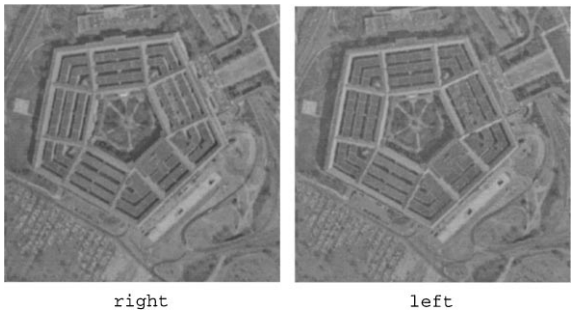

Figure 18. The Pentagon stereo pair.

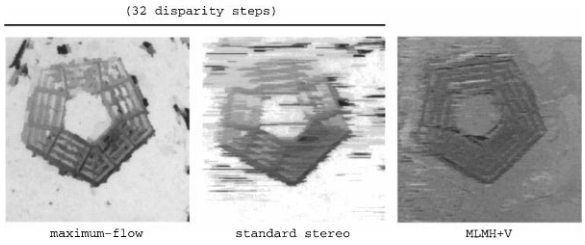

Figure 19. Disparity maps for the Pentagon stereo pair.

Figure 20. The Park meter stereo pair.

Figure 21. Disparity maps for the Park meter sequence. Results are shown for 2 image sequence.

Figure 22. Disparity maps for the Park meter. Results are shown for 4 image sequence. The matching volume is $256 \times 240 \times 64$.

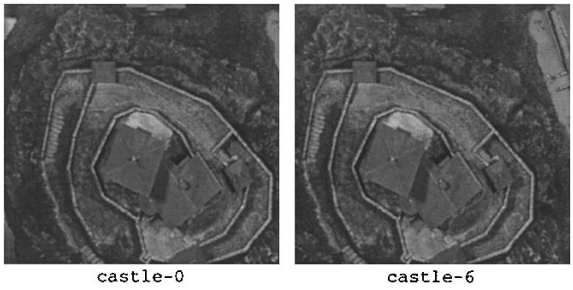

Figure 23. Two horizontally separated images from the sequence Roof.

of Section 4.1, affects the quality of the disparity maps. Figure 27 illustrates this for four levels of smoothness, namely $K = 0, 1, 10, 100$. For $K = 0$, the capacity of smoothness edges is zero and therefore each pixel is given a disparity independently of its neighbors. It is essentially equivalent to using direct

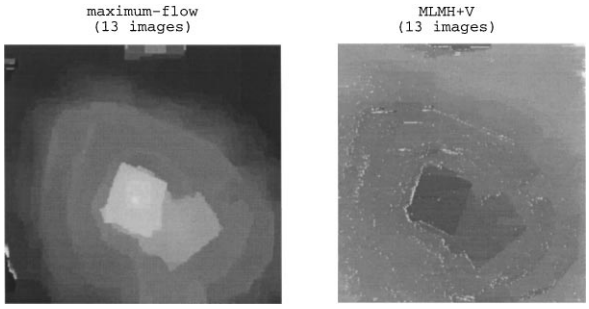

Figure 24. Disparity maps for the Roof sequence. Results are shown for 13 images. White points on the right denote detected occlusions. The maximum-flow matching volume is $256 \times 240 \times 64$.

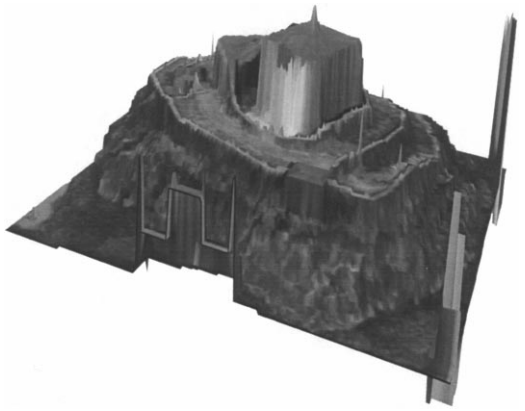

Figure 25. Reconstructed 3-d surface model for the Roof sequence. The depth map of maximum-flow disparity map is used.

search with correlation over a single pixel window (on the left of Fig. 27).

As expected, lowering the smoothness capacities favors depth discontinuities and therefore creates sharper object edges, at the expense of surface smoothness.

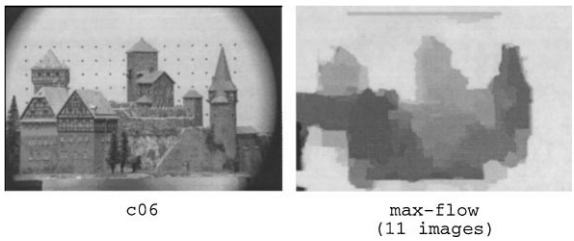

Figure 26. The Castle image stereo sequence. On the left, one of the 11 images. On the right, the resulting maximum-flow disparity map.

It is observed that large depth discontinuities tend to stay sharp as the level of smoothness increases. This is probably due to the fact that the smoothness is expressed in all directions instead of only along epipolar lines. This result differs strongly from most other methods where a high level of smoothness induces blurred or missing depth discontinuities.

6. Conclusion

We have presented a new algorithm for establishing n-camera stereo correspondence, based on a reformulation of the stereo matching problem to finding the maximum-flow in a graph. It is able to solve optimally for the full disparity surface in a single step, therefore avoiding the usual disparity inconsistencies across neighboring epipolar lines. The *ordering* constraint, required for dynamic programming, is replaced with a more general *local coherence* property that applies in all directions instead of along epipolar lines. The new stereo problem formulation supports multiple arbitrary cameras in a natural way and can estimate depth for an arbitrary virtual camera. Due to the global nature of the

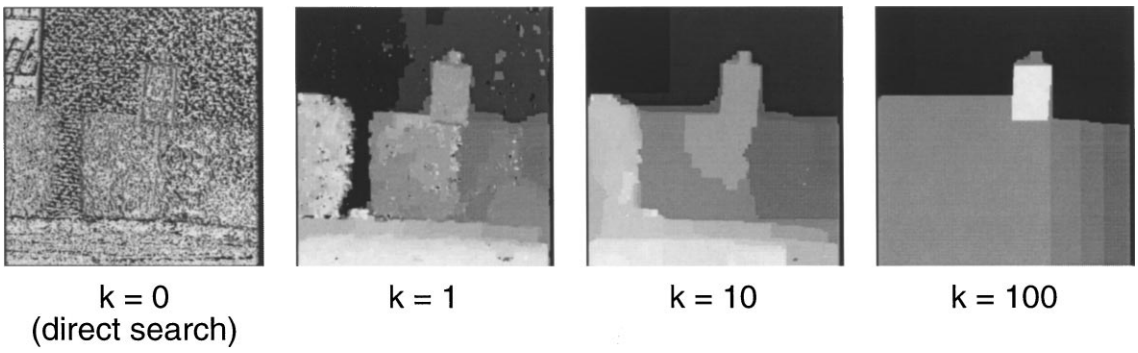

Figure 27. Disparity maps for the Shrub sequence for 4 smoothness levels. On the left, $K = 0$ enforce no smoothness. For $K = 1$, $K = 10$, and $K = 100$, progressively more smoothness is applied, resulting in graceful degradation of depth map.

minimization process, depth discontinuities are sharp and well localized, for any desired level of smoothness. Moreover, most spurious discontinuities are eliminated since smoothness is applied in all directions instead of only along epipolar lines.

We believe that this paper established clearly that a simple cost function, such as the one we used, can yield very high quality solutions when minimized globally and efficiently. These solutions rival and often surpass much more sophisticated cost functions that are impossible to globally minimize because of their complexity.

As for future research, there are many avenues open to improve the maximum-flow formulation proposed in this paper. In particular, a multi-resolution approach as well as local smoothness variations could be directly embedded in the graph, further improving performance and depth map quality.

Acknowledgment

I would like to thank Ingemar Cox, Jean Meunier and Neil Stewart for their suggestions and comments. I am grateful to Satish Rao and Andrew Goldberg for helpful discussions regarding the computation of maximum-flow in graphs.

Note

1. The nodes on the side of the graph are in fact less than 6-connected.

References

Baker, H.H. 1981. Depth from Edge and Intensity Based Stereo. Ph.D. Thesis. University of Illinois at Urbana-Champaign.

Belhumeur, P.N. 1996. A Bayesian approach to binocular stereopsis. *Int. J. Computer Vision*, 19(3):237–260.

Boykov, Y., Veksler, O., and Zabih, R. 1998. Markov random fields with efficient approximations. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA.

Cormen, T.H., Leiserson, C.E., and Rivest, R.L. 1990. *Introduction to Algorithms*. McGraw-Hill: New York.

Cox, I.J. 1994. A maximum likelihood N-camera stereo algorithm. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 733–739.

Cox, I.J., Hingorani, S., Maggs, B.M., and Rao, S.B. 1996. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567.

Faugeras, O. 1993. *Three-Dimentional Computer Vision*. MIT Press: Cambridge.

Goldberg, A.V. and Rao, S.B. 1997. Length functions for flow computations. Technical Report 97-055, NEC Research Institute, Princeton, NJ.

Greig, D.M., Porteous, B.T., and Seheult, A.H. 1989. Exact maximum a posteriori estimation for binary images. *J.R. Statist. Soc.*, 51(2):271–279.

Ishikawa, H. and Geiger, D. 1998. Segmentation by grouping junctions. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA.

Kanade, T., Yoshida, A., Oda, K., Kano, H., and Tanaka, M. 1996. A stereo machine for video-rate dense mapping and its new applications. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco.

Kang, S.B., Webb, J.A., Zitnick, C.L., and Kanade, T. 1994. An active multibaseline stereo system with real-time image acquisition. Technical Report CMU-CS-94-167, School of Computer Science, Carnegie Mellon University.

Marr, D. and Poggio, T. 1979. A theory of human stereopsis. *Proceedings of the Royal Society*, B 204:301–328.

Ohta, Y. and Kanade, T. 1985. Stereo by intra- and inter-scanline using dynamic programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7(2):139–154.

Roy, S. and Cox, I.J. 1998. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proc. Int. Conference on Computer Vision*, Bombay, India, pp. 492–499.

Yang, Y. and Yuille, A.L. 1995. Multilevel enhancement and detection of stereo disparity surfaces. *Artificial Intelligence*, 78:121–145.