PERGAMON

Pattern Recognition 34 (2001) 531–545

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Disparity maps for dynamic stereo☆

Gabriel Fielding[1], Moshe Kam*

Data Fusion Laboratory, Department of Electrical and Computer Engineering, Drexel University, 3141 Chestnut St., Philadelphia, PA 19104, USA

Received 11 March 1999; received in revised form 22 November 1999; accepted 22 November 1999

Abstract

Stereo matching algorithms often use regularization or relaxation methods to refine estimates of disparity in static images. Unfortunately, the computational requirements of these iterative techniques often preclude their use in real-time systems. Furthermore, most real-time stereo matching systems do not exploit the availability of a disparity map from the previous time step to compute the current disparity. We propose two algorithms for correspondence determination in dynamic stereo image sequences using prior disparity maps to localize, and to speed, the search for matches. The first method uses optical flow estimates in the stereo images to constrain the search space of feasible matchings. The second method uses a heuristic search scheme to prune matching graphs without explicit feature tracking or optical flow computation. Both methods can be applied to existing matching algorithms, to reduce their search space. The resulting improvement in matching is demonstrated in sample images and is quantified through the mean-square error of the computed disparity map compared to dense ground truth. The proposed methods are shown to improve the matching speed (decrease the latency) of five different matching algorithms. © 2001 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

1. Introduction

Stereo matching algorithms [1] use two or more camera images as inputs and provide a range or disparity map as output. Correlation-based stereo [2] provides dense correspondence by matching either single pixels or windows of pixels along corresponding epipolar lines. In this paper we propose two algorithms for correspondence determination in dynamic stereo image sequences. The first method uses optical flow to enhance exisiting matching algorithms. *Optical flow* is the apparent movement of brightness patterns in an image [3]. *Image flow* is defined as the 2D projection of the instantaneous 3D velocity of a point in the scene on the image plane [4]. Under the assumption that optical flow and image flow are equivalent [5], the optical flow fields in the left and right stereo cameras describe the trajectory of each pair of matched features or pixels in the stereo images.

Optical flow in calibrated stereo images has been used already to improve estimates of disparity. In Ref. [6], Waxman and Duncan incorporated the components of optical flow in parallel stereo cameras into a similarity function used to match features in the images. Later, Duncan et al. [7] estimated the 3D motion of an observer by recovering both optical flow and correspondence from a set of features in the stereo images. Matthies et al. [8] developed a method of extracting depth from camera motion assuming the camera motion is known (this approach cannot be used with moving objects in the image sequence).

In this study we take a different approach to integrating optical flow information into a stereo matching algorithm. We create *search filters* that use optical flow information and a prior disparity map to constrain the

☆This research sponsored by an National Defense Science and Engineering Graduate research fellowship from the Office of Naval Research and by the National Science Foundation under grants ECS 9057587, ECS 9216588, and ECS 9512363.

[1] Currently with the Imaging Science Technology Labs, Eastman Kodak Company, 1700 Dewey Avenue, Rochester, NY 14650, USA.

* Corresponding author. Tel.: + 1-215-895-6920; fax: + 1-215-895-1695.

E-mail address: kam@minerva.ece.drexel.edu (M. Kam).

PII: S0031-3203(00)00011-X

search space of candidate matches (i.e., prune the search graph and increase the speed of finding a path or matching). Previous approaches required either that the observer motion be known or that it be accurately estimated from the flow field before the flow estimate is used to refine disparity. Unfortunately, precise observer motion is rarely available, and recovering observer motion from the optical flow is problematic if moving obstacles are not subtracted from the background motion [7]. The methods presented here allow the scene to have multiple moving objects and do not require knowledge of observer motion. Moreover, they do not require the flow fields in different cameras to be precise for corresponding pixels in the stereo image.

Assuming that the dense optical flow field is available (along with an uncertainty measure for each flow estimate), we propose that the optical flow field be used to improve the stereo matching process. The flow vector acts as a pointer to a previous estimate of disparity which suggests a region to consider in the matching graph.

If optical flow cannot be used (due to the high price of computation or excessive noise) we propose an alternative. Assuming that the amount of inter-frame motion (in pixels) is bounded by some known constant, we can use a local window of disparity estimates, calculated from the previous disparity map, to restrict the search space. In other words, rather than using optical flow to propagate particular disparity estimates forward frame by frame, *all* prior disparities within a local neighborhood of a pixel are used to constrain the search.

The paper is arranged in three main sections. In Section 2, we describe the two main areas that this paper draws from, namely stereo matching and optical flow. In Section 3, we describe two new methods for integrating optical flow with stereo matching algorithms. Finally in Section 4 we give a detailed performance analysis on synthetic and real stereo image sequences.

2. Preliminaries

In this section, we introduce the notation used throughout the paper. We also briefly describe *stereo matching* and *optical flow*. The description is intended to provide those unfamiliar with these areas with enough background to understand our contributions in Section 3. Readers already familiar with these subjects may wish to skip this section and continue directly to Section 3.

An image is denoted $I_s(x_s, y_s, t) \in \mathbf{R}$ where x_s and y_s are the spatial coordinates in the image, t is the time at which the image was taken and s is a label for each distinct camera. Here, we have two cameras with $s = l$ for 'left' and $s = r$ for 'right'. We use discrete time, $t_k = k\Delta t$ where Δt is a fixed interval between image frames and k is an integer identifying the frame. When the time instant is clear from the context, we will denote an image by $I_s(x_s, y_s)$. The *disparity map* $D_s(x_s, y_s, t) \in \mathbf{R}$ is an array whose elements specify the offset of a pixel in one image to its "match" in another image, for any pair of images. This means that given pixel (x_r, y_r) in the right image, the matching pixel in the left image is located at pixel $(x_l, y_l) = (x_r + D_r(x_r, y_r, t), y_r)$; that is, $x_l = x_r + D_r(x_r, y, t)$. $D_s(x_s, y_s, t)$ is undefined for pixels which have no match. All matching algorithms in this paper compute integer-valued disparities (i.e., $D_s(x_s, y_s, t) \in \mathbf{I}$) on discretely sampled images. The true disparity is a real number (which may require interpolation techniques to refine the *precision* of the integer estimate). In order to distinguish between iterative disparity refinements from static images and dynamic disparity maps, we denote a disparity map at iteration j by $D_s^j(x_s, y_s, t)$.

2.1. Stereo correspondence

Fig. 1 shows the image planes of two cameras in a parallel stereo configuration. The lenses are represented by points $\mathbf{f}_\mathrm{l}$ and $\mathbf{f}_\mathrm{r}$ which are drawn behind the image planes to simplify the projections. Given some arbitrary point on a visible surface in the scene, say point $\mathbf{Q}$, the triplet of points, $\mathbf{f}_\mathrm{r}$, $\mathbf{Q}$ and $\mathbf{f}_\mathrm{l}$ defines a unique plane. The intersection of this plane with the left image and right image defines two lines which are known as *epipolar lines*. The epipolar lines represent the possible locations of corresponding projections. Hence, given two corresponding epipolar lines, we attempt to determine the correspondence for each point on the two lines. Given a point on any one epipolar line, this *epipolar constraint* reduces the search for the matching point from two dimensions to one dimension. We assume that the cameras are calibrated (e.g. Ref. [9]) and the corresponding epipolar lines are known through this calibration process. We say the epipolar lines are *registered* under this assumption, and we can tranform the stereo image data such that $y_r = y_l = y$.

Occlusions occur when a surface obscures or hides another surface in space from a particular viewing

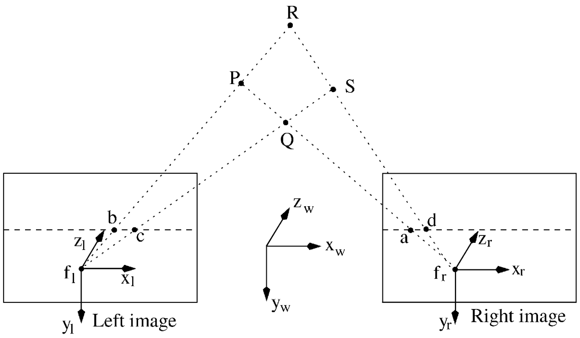


Fig. 1. Epipolar lines in a parallel stereo configuration.

location. In Fig. 1 this might occur when there exists some surface passing through point **P** and another surface through point **R**. Both points on the surfaces are visible in the right image but only **P** is visible in the left image.

With each potential matching (x_l, x_r) on corresponding epipolar lines we associate a similarity cost $c(x_l, x_r) \in [0, C_M]$ where C_M is an upper bound on the similarity function. We use the sum-of-squared differences (SSD) as the measure of similarity between two square windows. Given some time instant t and a specific epipolar line y, SSD is defined by

$$c(x_l, x_r) = \sum_{-p \leqslant i,j \leqslant p} [I_r(x_r + i, y + j, t) - I_l(x_l + i, y + j, t)]^2 \quad (1)$$

for some integer $p \geqslant 0$ that controls the window size. The values of y and t are the same for both images. To match x_l and x_r, one typically minimizes some energy function based on Eq. (1). We define the *disparity* as the difference between the x-coordinates in the stereo images, $d = x_l - x_r$. It can be shown that the disparity is inversely proportional to the distance of the object viewed in the stereo images [10].

For points that are occluded, all potential matches will usually yield large values of $c(x_l, x_r)$. In Ref. [11], Cox et al. describe a method and algorithm for computing disparities and identifying occlusions. This method was expanded in Ref. [12] and in Appendix B.

2.2. Matching algorithms

A *matching* is a set of pairings of pixels in the left and right images: $\{(x_l, x_r)\}$. From this matching set we construct the disparity map in each image, $D_r(x_r, y, t) = x_l - x_r$ and $D_l(x_l, y, t) = x_r - x_l$.

There are several algorithms to solve the correspondence problem in stereo images along registered epipolar lines (e.g. Refs. [11,13,14]). Most algorithms solve an equivalent graph search problem. The two most common graph representations are given in Figs. 2 and 3. Fig. 2 shows a weighted graph with pixels from the left and right epipolar lines drawn as nodes (circles) and potential matches between those pixels drawn as edges (lines). Each edge is associated with a weight computed through a similarity measure. Fig. 3 shows a weighted directed graph where the weight on each edge is considered a distance. Thus, the shortest path between the vertices labeled S and T can be used to assign stereo correspondences.

In Appendix A (and in Ref. [1]), we describe five matching algorithms that use the similarity values in Eq. (1) and an occlusion threshold C_O to produce a disparity map. Each of the five algorithms is based on a distinct optimization technique, namely: (i) *local search*, (ii) the *left–right heuristic* (a heuristic agreement algorithm), (iii) *dynamic programming*, (iv) *maximum-weighted bipartite matching*, and (v) *greedy matching*. We shall show that our two search localization methods can enhance all five algorithms.

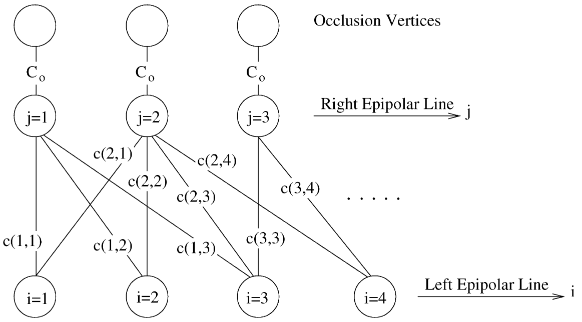


Fig. 2. A representation of the correspondence problem as a weighted graph where weights are given by $c(x_l, x_r)$ and the occlusion costs are given by C_O.

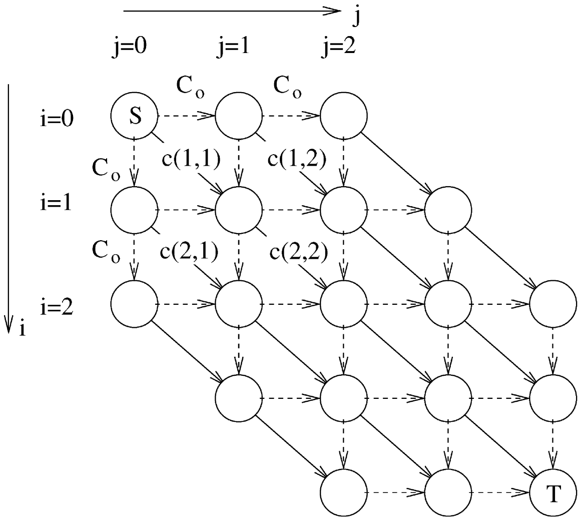


Fig. 3. A graph representation of the stereo matching problem where the similarity between pixels is represented as a weight on the diagonal edges of the graph. The dotted edges allow for occlusions to occur and their weights are given by the *occlusion cost*, C_O.

2.3. Optical flow

Optical flow is the apparent movement of brightness patterns in an image [3]. *Image flow* is defined as the 2D projection of the instantaneous 3D velocity of a point in the scene on the image plane [4]. Motion of objects and/or motion of the camera produces image flow as

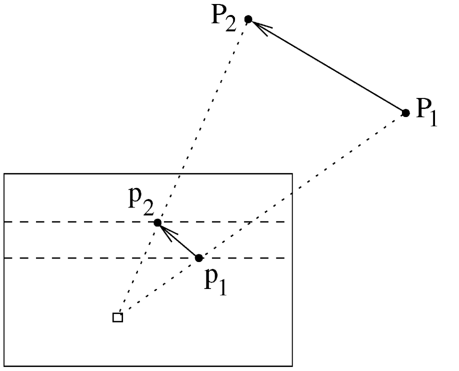


Fig. 4. The motion of a point $P_1(x_w, y_w, z_w, t_1)$ in world coordinates to a point $P_2(\tilde{x}_w, \tilde{y}_w, \tilde{z}_w, t_2)$ results in image flow, the change of the point's projection from $p_1(x, y, t_1)$ to $p_2(\tilde{x}, \tilde{y}, t_2)$.

shown in Fig. 4 where we depict a point P_1 and a vector indicating the motion of that point to some other point P_2 at a later time. The change in location of the underlying surface, from P_1 to P_2, results in a change in the projection of the surface in the image plane, from p_1 to p_2.

Image flow is distinct from optical flow. One example of the difference between an optical flow and an image flow occurs when the lights in a scene are being dimmed: the brightness patterns change (so there is an optical flow) but there is no image flow. It can be shown that under certain conditions, including spatio-temporally uniform lighting [4], the component of the image flow normal to the brightness gradient is equal to the component of the optical flow normal to the brightness gradient. In general, however, the two fields are distinct. In the sequel, we use the optical flow as though it were image flow; hence, we make the common assumption that the two flow fields are equivalent [6].

Dense optical flow can be obtained through gradient approaches coupled with smoothness constraints [3] or through correlation-based matching [15]. We obtain dense optical flow for every pixel using the multi-resolution correlation-based matching approach in Ref. [16]. With dense optical flow fields, we can develop a *flow-based graph pruning* scheme to refine disparity estimates.

Following [3], we assume that we are given a camera s and an image frame, $I_s(x_s, y_s, t)$, and that the brightness pattern remains constant over time. Under these circumstances,

$$I_s(x_s, y_s, t) = I_s(x_s + \delta x, y_s + \delta y, t + \delta t),$$

where δx_s, δy_s, and δt are differentials with respect to x_s, y_s, and t, respectively. Taylor-series expansion of the right-hand side up to the first-order terms results in

$$\delta x_s \frac{I_s}{x_s} + \delta y_s \frac{I_s}{y_s} + \delta t \frac{I_s}{t} = 0.$$

By dividing through by δt and setting $(u, v) = (\delta x_s/\delta t, \delta y_s/\delta t)$, we obtain

$$\left(\frac{\partial I_s}{\partial x_s}, \frac{\partial I_s}{\partial y_s}\right)(u, v) = -\frac{\partial I_s}{\partial t}, \quad (2)$$

where (u, v) represents the optical flow at pixel (x_s, y_s) at time t. We denote the dense set of optical flow estimates through the pair of arrays $(U(x_s, y_s, t), V(x_s, y_s, t))$, where U is the horizontal component of optical flow and V is the vertical component of optical flow. Since in Eq. (2) we have one equation with two unknowns, we cannot recover the optical flow uniquely. This ambiguity in the gradient approach is known as the *aperture problem* which states that we can recover only the normal component of the optical flow [3], also called the *normal flow*. Correlation-based matching approaches also suffer from this problem.

3. Constraining the search

In this section we first describe several typical methods for improving the accuracy of stereo matches. These refinement methods are iterative methods performed off-line. If new images become available, it is desirable to compute stereo matches for the current frames using unrefined stereo matches from the previous time step. To this end, we describe in Sections 3.1 and 3.2 two algorithms for online disparity refinement.

Smoothness constraints and penalty terms for static images: The most prevalent means of refining a disparity map is through some form of relaxation or regularization [17–19]. By refinement we mean the improvement of the estimated disparity for a given set of stereo images through additional filtering. This usually involves the use of a penalty term to enforce a "smoothness constraint". The constraint assumes that surfaces in space are generally smooth and should have slowly changing disparity maps (spatially). One form of the penalty term[2] [2] is based on depth gradient

$$\text{penalty} = \lambda|\nabla \hat{D}_s^{j-1}(x_s, y_s, t)|^2. \quad (3)$$

Here λ is a scalar which controls the amount of smoothing, ∇ is the gradient operator, and $\hat{D}_s^{j-1}$ is the prior disparity map (namely, the map at iteration $j-1$) with the previous disparity estimate at pixel (x_s, y_s) replaced with the candidate disparity estimate. An additional example of a smoothness constraint is available in Ref. [20].

Map refining for dynamic images: Penalty terms are attractive for iterative refinement of the disparity map for

[2] For additional examples see Ref. [20].

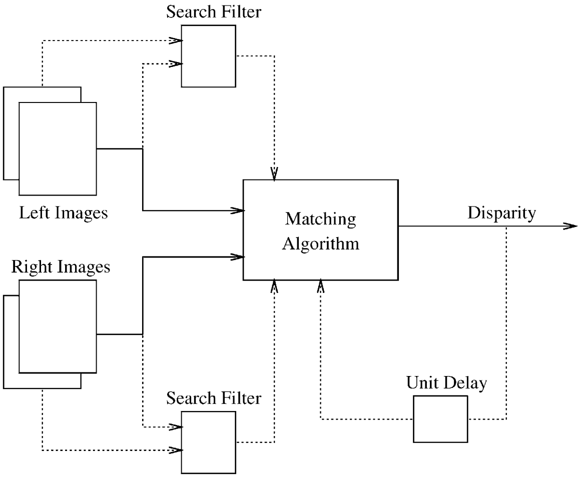


Fig. 5. A block diagram of the matching process studied in this paper. For the search filter we use either (i) the optical flow fields which constrain the search space from prior disparity maps (Section 3.1); or (ii) a weighting function based on optical flow computations which reweights a fixed local neighborhood of prior disparity estimates (Section 3.2).

static image pairs. Unfortunately, this approach cannot be extended directly to incorporate prior disparity maps when the scene is changing. When the input is dynamic, the penalty term is applied erroneously to regions that have changed due to moving obstacles or moving cameras.

In Fig. 5 we show a block diagram of the matching processes studied here. In this diagram, the inputs to the matching system are the raw image sequences from the two cameras. The output is the disparity map, $D_s(x_s, y_s, t)$. The blocks labeled *search filter* take the current image $I_s(x_s, y_s, t)$ and the previous image $I_s(x_s, y_s, t - \Delta t)$ and produce information to guide the search process in the matching block.

We present two implementations of the search filter. In Section 3.1 the search filter computes optical flow vectors for a dense set of points in the current image. In Section 3.2 the search filter computes a vector of weights over a local neighborhood at each image pixel. In both cases, the data from the search filter are passed to the matching algorithm, where they are used to prune the search graph.

3.1. Graph pruning using optical flow estimates

Given two frames taken Δt apart, $I_s(x_s, y_s, t - \Delta t)$ and $I_s(x_s, y_s, t)$, the vertical and horizontal components of the optical flow field in each camera are given by $U_s(x_s, y_s, t - \Delta t)$ and $V_s(x_s, y_s, t - \Delta t)$. These describe the flow field *from* the previous frame *to* the current frame. We can also compute the *reverse* flow *from* the current frame *to* the previous frame $U'_s(x_s, y_s, t)$ and $V'_s(x_s, y_s, t)$ by reversing the order of the two frames. We define $u'_s(x_s, y_s) = U'_s(x_s, y_s, t)$ and $v'_s(x_s, y_s) = V'_s(x_s, y_s, t)$ to be the components of reverse optical flow at the current time.

For a given pixel in an image $p_s = (x_s, y_s)$, we define a set of prior disparity estimates using the optical flow vectors in a local window around p_s,

$$\Omega_s = \{D_s(x_s + i + u'_s(x_s + i, y_s + j), y_s + j + v'_s(x_s + i, y_s + j), t - \Delta t) \colon i, j \in W\}. \tag{4}$$

Here W is a window around the selected pixel (x_s, y_s), given by $\sqrt{i^2 + j^2} \leqslant R$ for some radius R chosen based on the uncertainty of the optical flow estimate at p_s.[3] In the absence of confidence measures for the optical flow, a window of fixed radius can be used. Since the surface whose projection lies at (x_s, y_s) may be moving, we include a range of disparities about each estimate in Ω_s to account for motion away from or towards the observer. We introduce the parameter ε to indicate an interval of disparities around each element of Ω_s within which to search for a match. Since disparity is inversely proportional to range, z_w, in parallel stereo cameras, bounds on the derivative $\dot{z}_w$ can be used to derive a bound on $\dot{d}$ (see Appendix C) and thus on ε. In this work, we use the following function of the prior disparity estimate:

$$\varepsilon_{ij} = \left\lceil k \left| \frac{\hat{d}_{ij}}{d_{MAX}} \right| + 0.5 \right\rceil, \quad \hat{d}_{ij} \in \Omega_s, \tag{5}$$

where $\hat{d}_{ij}$ denotes a prior disparity estimate (an element in Ω_s, Eq. (4)), d_{MAX} is the maximum disparity one intends to recover, and $k \geqslant 0$ is a scalar parameter controlling the maximum change of disparity values for a single time step.[4] Using the definition of ε_{ij} we can define an *augmented neighborhood* of disparities for each $\hat{d}_{ij}$,

$$\phi_{ij} = \{\xi \colon \hat{d}_{ij} - \varepsilon_{ij} \leqslant \xi \leqslant \hat{d}_{ij} + \varepsilon_{ij}, \quad \hat{d}_{ij} \in \Omega_s\} \tag{6}$$

and for each element in the augmented neighborhood we have a weighting function,[5]

$$\psi_{ij}(\xi) = \exp\left\{ -\frac{(\xi - \hat{d}_{ij})^2}{\beta^2} \right\}, \quad \forall \xi \in \phi_{ij}, \tag{7}$$

[3] Some optical flow estimates [4] provide the uncertainty in the vertical and horizontal direction; others [16] estimate uncertainty along the pricipal axis of curvature of the matching surface.

[4] In Eq. (5), we use the ceiling function for discrete matching without interpolation. ceil(x) = $\lceil x \rceil$ is the smallest integer greater than or equal to x.

[5] Alternative weighting functions incorporating the uncertainty of the prior disparity (e.g. Ref. [21]) as a shaping parameter are possible.

where β is a shaping parameter that controls the rate at which disparity may change between frames. Small values of β create a sharp rolloff in the weighting function, resulting in less change in disparities between frames. Large values of β let the weights computed from Eq. (1) dominate, which allows disparity transitions to occur more freely. In the remainder of this work we set $k = 5$ and $\beta = 5.0$.

The total range of disparities to be searched is given through the union of the augmented neighborhoods

$$\Phi_s = \bigcup_{i,j \in W} \phi_{ij}. \tag{8}$$

We retain the largest weight for each estimate at $\xi \in \Phi_s$

$$\Psi_s(\xi) = \max_{i,j \in W} (\psi_{ij}(\xi)). \tag{9}$$

For a given pair of epipolar lines, Eqs. (4)–(9) are computed for each element on the left and right epipolar lines. A weighting matrix $w(x_l, x_r)$ is given through the following:

$$w_r(x_r + \xi, x_r) = \Psi_r(\xi), \quad \forall \xi \in \Phi_r,$$

$$w_l(x_l, x_l - \xi) = \Psi_l(\xi), \quad \forall \xi \in \Phi_l,$$

$$w(x_l, x_r) = \max(w_l(x_l, x_r), w_r(x_l, x_r)). \tag{10}$$

The weighting matrix in Eq. (10) is used to reweight the similarity matrix in Eq. (1):

$$\hat{c}(x_l, x_r) = C_M(1 - w(x_l, x_r)) + w(x_l, x_r)c(x_l, x_r).$$

Prior disparity maps may include occluded regions. Hence, some of the prior estimates in Eq. (4) will be labeled occlusions. For these cases, we cannot use Eq. (8) to describe an augmented disparity range for the search. Instead, we define the following rule for occlusions: if less than half of the prior disparities in Eq. (4) are occlusions, then we ignore the occlusions and use the augmented estimates in Eq. (8); if more than half of the prior disparity estimates are occlusions, then we revert to the case where no prior disparity map is given. That is, instead of Eq. (8) we use

$$\Phi_s = \{\xi : d_{MIN} \leqslant \xi \leqslant d_{MAX}\}, \quad \text{and} \quad \Psi_s(\xi) = 1 \ \forall \xi \in \Phi_s, \tag{11}$$

with d_{MIN} being the minimum disparity one desires to recover (usually zero in parallel stereo systems). This rule, coupled with the use of an occlusion threshold [1,11], allows objects to be detected at the edges of the stereo images as the camera pans around or as objects move into view. Moreover, this rule can prevent disparity errors from accumulating over many frames. We have observed that incorrect prior estimates eventually cause several false occlusion declarations. This in turn causes the subsequent refinement iteration to use Eq. (11) rather than Eq. (8) allowing for correction of these false occlusion declarations.

Typically, the augmented disparities in Eq. (8) represent a fraction of the total range of possible disparities. This means that use of Eq. (8) reduces the space of feasible matches. In terms of the matching graphs in Figs. 2 and 3, the algorithm can be viewed as a pruning operation. For a given prior disparity estimate $\hat{d} = D_r(x_r, y_r, t - \Delta t)$, only pairs within a range of ε of $\hat{d}$ are included in the graph. Thus only $c(x_r, x_r + \xi)$, $\forall \xi \in \Phi_r$ are included in the graph. Furthermore, the weights of these included edges are modified according to Eq. (7). When Eq. (11) is used due to occlusions in the prior disparity map, the matching graph is not pruned at all.

3.2. Graph pruning using fixed local neighborhoods

In the previous section we used reverse optical flow to guide the search process in computing disparity. Most optical flow algorithms make assumptions on the amount of inter-frame motion to constrain the computations. If this inter-frame motion is only a few pixels, we propose a *fixed local neighborhood* scheme to refine disparity estimates.

Rather than using flow explicitly to select previous disparity estimates, we define a local neighborhood around a pixel in the current image and use that same window in the previous disparity map to define a list of candidate disparities:

$$\Omega_s = \{D_s(x_s + i, y_s + j, t - \Delta t) : i, j \in W\}. \tag{12}$$

Here W is a window around the selected pixel (x_s, y_s) as in Eq. (4). The scalar parameter ε in Eq. (5) remains the same as does the augmented neighborhood in Eq. (8) and the weighting function in Eq. (7). For each element in Eq. (8) we define an additional weighting function based on the similarity between two windows centered at $I_s(x_s + i, y_s + j, t - \Delta t)$ and $I_s(x_s, y_s, t)$, where (i,j) represents the offset from the pixel of interest in our fixed local neighborhood. This weighting function is

$$\varphi_{ij} = \exp\left\{ - \sum_{-p \leqslant \zeta, \eta \leqslant p} [I_s(x_s + \zeta, y_s + \eta, t) - I_s(x_s + i + \zeta, y_s + j + \eta, t - \Delta t)]^2 \right\}. \tag{13}$$

This weighting function is computed once for each element in Eq. (12) and is applied to each element in the appropriate augmented neighborhood in Eq. (6). Thus, we modify Eq. (9) to include the additional weighting function as follows:

$$\Psi_s(\xi) = \max_{i,j \in W} (\varphi_{ij}(\xi)\psi_{ij}(\xi)). \tag{14}$$

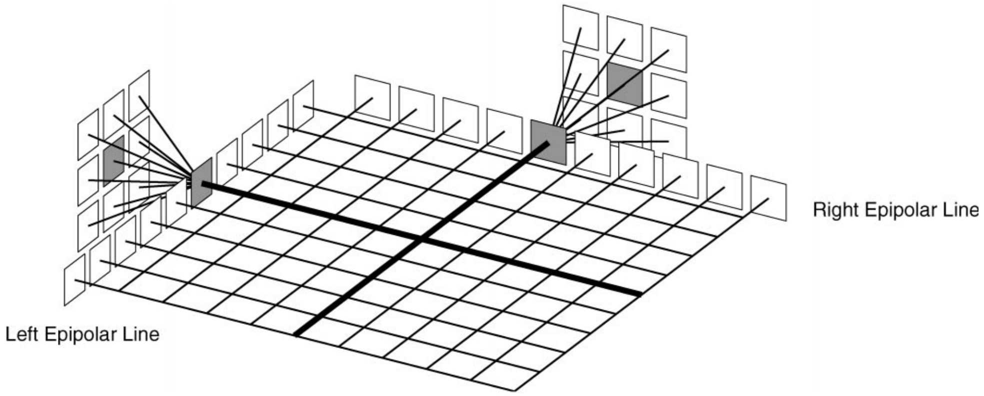


Fig. 6. The prior disparity information is incorporated into the epipolar cost matrix using fixed local neighborhoods. The dark lines show the row and column of the epipolar cost matrix affected by pruning and reweighting. For a given pair of epipolar lines, both refinement schemes provide a weighted set of prior disparities that are used in the epipolar cost matrix.

The remainder of the reweighting scheme remains identical to that described previously. When there is no optical flow — because the camera is stationary, there are no moving objects, and lighting is constant — Ω_s in Eqs. (4) and (12) are equivalent and all the weights in Eq. (13) become unity.

In Fig. 6 we show how the prior disparity information is incorporated into the epipolar cost matrix using fixed local neighborhoods. Once the reweighting has been done on the epipolar cost matrix, $c(x_l, x_r)$, a matching algorithm computes the disparities.

Both disparity refinement schemes are typically unaffected by flow errors caused by the "aperture problem" (see Section 2.3). For the optical-flow-based algorithm, when the optical flow is computed, the normal flow vector will usually indicate a point on the same underlying surface with the same disparity. The fixed local neighborhoods introduce the weighting function in Eq. (13), which produces identical weightings for indistinguishable matches. However, the correct choice is still present in the epipolar matching graph and the ambiguity may be resolved during the stereo matching phase.

4. Performance analysis

Along the top row (a) of Fig. 7 we show three frames from the right camera of a synthetic stereo image sequence where each image is grayscale and 256 pixels square. The observer (i.e., cameras) is moving forward until frame 30, when it pans right. We searched over a disparity range of 100 pixels. All stereo matching for this example used the greedy matching algorithm in Ref. [1].

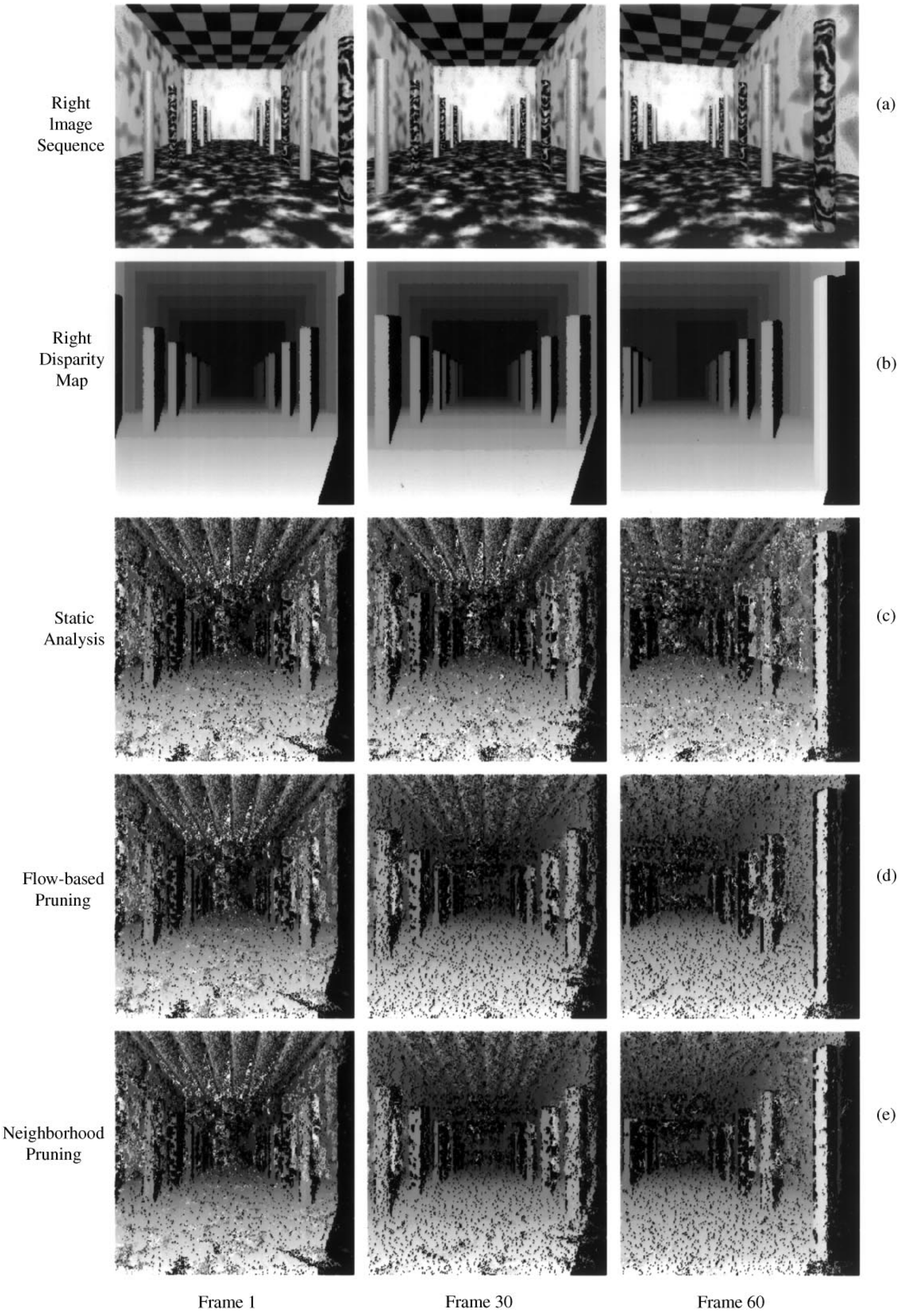
Fig. 7(b) shows three frames of the true disparity map with occluded regions marked in black. Larger disparities (closer objects) are represented by lighter colors.

The images in the third row (c) of Fig. 7 show the static stereo analysis where each new stereo pair is considered independently of previous frames.

The analyses using the refinement schemes in this paper are shown in rows (d) and (e) of Fig. 7. The analysis according to the flow-based pruning scheme is given along row (d). The improvement through this dynamic scheme is visually perceptible as well as quantifiable through the mean-square-error (MSE) values. We show the results of the local neighborhood pruning method along row (e) in Fig. 7. Visually the images are very similar to those from the flow-based method.

The MSE versus frame number for the image sequences in rows (c), (d), and (e) of Fig. 7 are given in Figs. 8, 9 and 10, respectively. These show the MSE for static analysis, flow-based pruning, and fixed local neighborhood analysis. All matching algorithms achieve improved performance over static analysis using the refinement strategies presented in this paper. Of particular interest is the very similar performance of the flow-based refinement strategy and the local-neighborhood refinement strategy. In fact, local-neighborhood refinement is marginally superior to flow-based refinement for every underlying matching algorithm except for dynamic programming. While DP provides a very good performance in terms of MSE for the static case, the algorithm produces matching artifacts because of the ordering constraint (see discussion in Ref. [1]).

An interesting characteristic of the refinement strategies' performances, shown in Figs. 9 and 10, is the apparent convergence of the error values for all five matching techniques. While we cannot prove the convergence of the algorithms we have described, our experience is that the performance *always* improved with the use of our two refinement algorithms. An iterative refinement strategy that used disparity gradient constraints


Right Image Sequence
(a)
Right Disparity Map
(b)
Static Analysis
(c)
Flow-based Pruning
(d)
Neighborhood Pruning
(e)
Frame 1
Frame 30
Frame 60

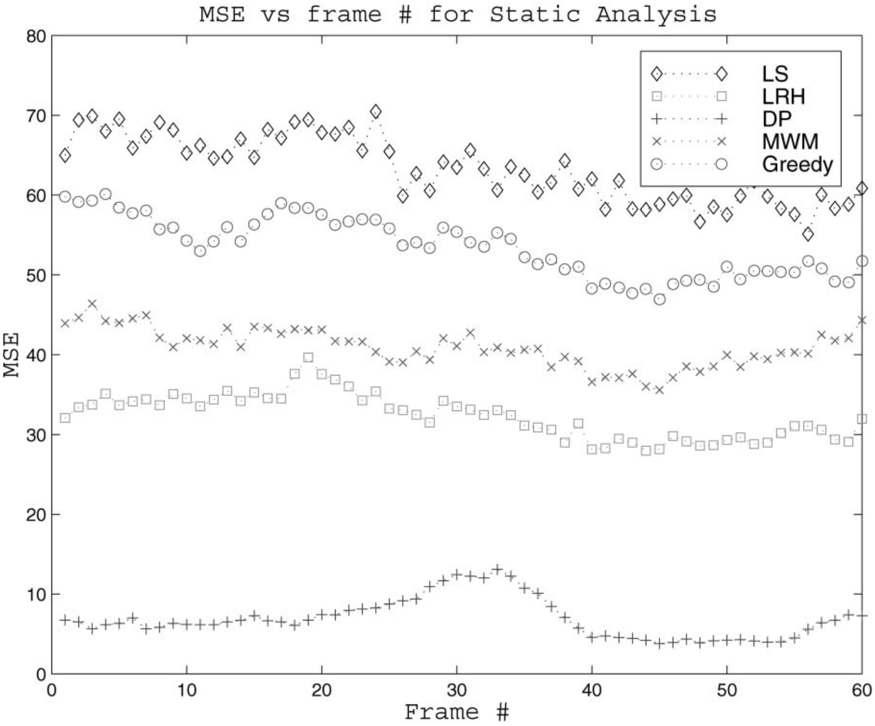


Fig. 8. MSE for static analysis presented along the row (c) of Fig. 7. No improvement occurs as the sequence proceeds frame by frame because prior information is not used.

showed a similar convergence behavior. However, our system is converging while integrating new stereo images. This is one of the main advantages of our techniques when compared to offline disparity refinement strategies. (An obvious consideration is whether one can accept the initialization period when the disparity maps tend to have larger errors.)

In Table 1 we give the running times for the five algorithms with the two refinement schemes as well as a static analysis. The times in the table are given relative to the static version of the local search matching (given a value of 1.0 in the upper left cell of the table). The times are averages over the 60 frame sequences. The flow-based scheme's running time is separated into the constant time required for the optical flow algorithm and the time required by the matching algorithm on the pruned graph.[6] All times were computed on a Sun Enterprise 450 with a 248 MHz SUNW, UltraSPARC-II processor. A rating of 1.0 requires approximately 6 s of CPU time. The refinement strategies offer significant speed improvements because of the reduced search space after pruning. One interesting observation is that while maximum weighted matchings tend to offer superior MSE performance over greedy matchings in the static case (see Fig. 8), this performance gap closed once a dynamic refinement strategy was added (Figs. 9 and 10). This fact coupled with the running times in Table 1 suggest that the greedy matching algorithm can be the best matching algorithm when narrow occluding objects are present.

To demonstrate the two refinement strategies further, we have analyzed the real scene in Fig. 11 (row (a)) where the image value at each pixel was corrupted by additive

◀

Fig. 7. The right image from a synthetic stereo image sequence is shown along the top row (a) and the true disparity map with respect to the right image is shown in row (b) where larger disparities (closer objects) are represented by lighter colors, and occluded regions are marked in black. The camera is moving forward until frame 30 at which point it pans right. The static analysis along row (c) considers every stereo pair independently of the previous frames. The flow-based analysis along row (d) uses prior disparity estimates pointed to by the optical flow in a local window to prune the current search graph. The bottom row (e) shows the results of a pruning/reweighting scheme using a fixed local neighborhood.

[6] This separation of computing time into *optical flow* and *matching* is given because recent hardware implementations of dense flow estimation (e.g. Ref. [22]) may provide flow estimates at low computational costs.

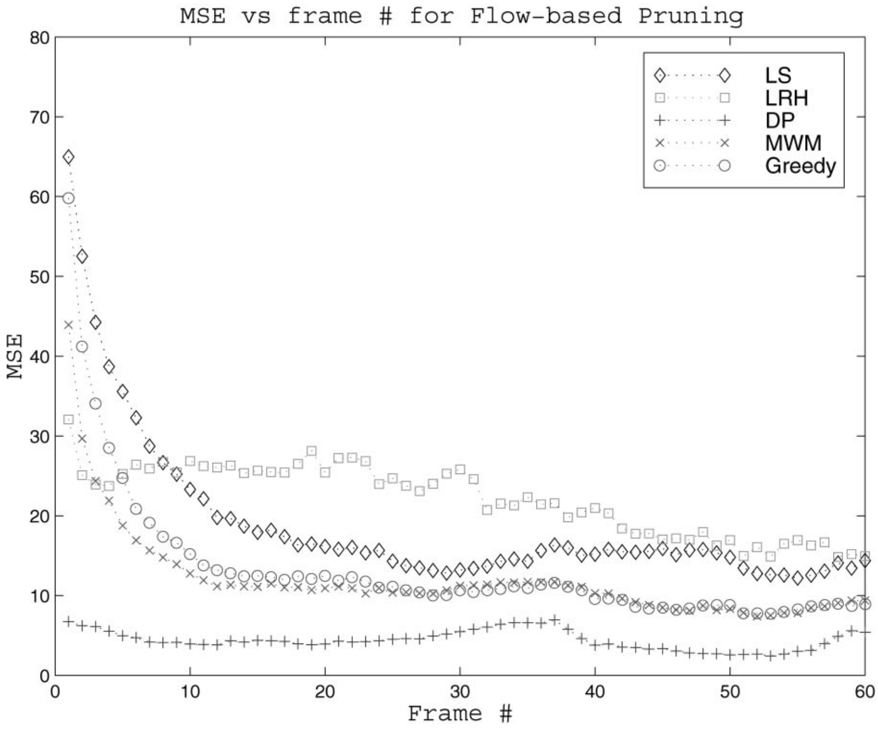


Fig. 9. MSE for flow-based refinement strategy presented along row (d) of Fig. 7. Very clear improvement occurs as the sequence proceeds because prior disparity information is used.

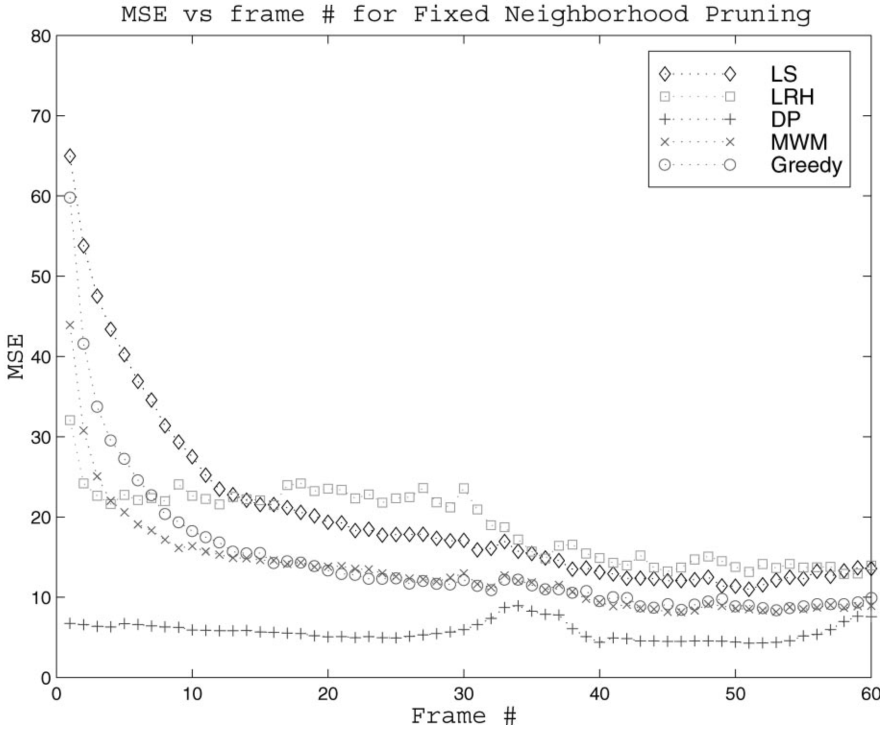


Fig. 10. MSE for local-neighborhood refinement strategy presented along row (e) of Fig. 7. The gradual improvement of the disparity information is clear.

Table 1
Average running times for the five algorithms and the refinement schemes for the 60 frames of the synthetic test sequence in Fig. 7. Times are given in units of the time needed by the local search algorithm assuming a static image. The flow-based refinement algorithm assumes that the flow is precomputed (requiring 7.21 units of time)

	Static	Flow-based		Local neighborhoods
		Optical flow	Matching	
Local search (LS)	1.00	7.21	0.34	0.44
Left-right heuristic (LRH)	1.07	7.21	0.54	0.68
Dynamic programming (DP)	1.31	7.21	0.61	0.74
Maximum weighted matching (MWM)	12.1	7.21	8.30	9.80
Greedy matching (GM)	1.44	7.21	0.71	0.84

Gaussian white noise with zero mean and $\sigma = 5$. The left image (not shown) was translated one pixel horizontally and one pixel vertically in the second frame, and rotated by one degree in the third frame to simulate camera misalignment. We show the results of static analyses and the two proposed search filtering strategies in Fig. 11 using the greedy matching algorithm. Along row (b) of Fig. 11 we show three frames analyzed without using prior information. Row (c) uses the flow-based refinement strategy to prune the epipolar cost matrix and then uses the greedy algorithm to select matches. Row (d) shows how the local neighborhood algorithm slowly refines the disparity estimates with similar performance to the explicit flow-based algorithm. Both refinement strategies succeed in improving the disparity map in the presence of camera misalignment and noise.

5. Conclusion

We have presented two methods for using a prior disparity map with optical flow to prune the graph representing feasible matchings. The first method uses optical flow estimates to guide the search for reliable prior estimates of disparity. The second method uses a local window of prior estimates to prune the matching graph. Both techniques can be integrated into existing matching algorithms to refine stereo matches online even in the presence of moving obstacles and camera movement.

The algorithms we described may run in an online fashion to provide improved disparity maps. This does not preclude the use of traditional offline refinement methods. Indeed, if sufficient computation is available then it would be desirable to have both on-line and off-line disparity refinement. Often, one must trade off image resolution (and hence, depth resolution) with speed requirements. Having this online refinement method will be especially useful for those trying to achieve the right balance of resolution and speed.

Appendix A. Matching algorithms

In Ref. [1], we describe five matching algorithms that use an array of similarity values as in Eq. (1) and a threshold C_O to produce a dense disparity map. Each of the five algorithms is based on a distinct optimization technique: (i) *local search*, (ii) *left–right heuristic* (a heuristic agreement algorithm), (iii) *dynamic programming*, (iv) *maximum-weighted bipartite matching*, and (v) *greedy matching*. We summarize the five algorithms here.

The *epipolar cost matrix*, $c(x_l, x_r)$ describes the cost of matching a point x_l with a point x_r. We assume that the values are defined by some function which is non-negative and for which the minima of the function occur at points that match. Given an upper bound, C_M, on this matrix we define the alternative matrix $a(x_l, x_r) = C_M - c(x_l, x_r)$ where matches are represented by maxima.

A.1. Local search [13,23]

A widely used approach for determining disparity is to compute the values of a similarity function for each candidate pixel and to choose the disparity that corresponds to the minimum of that function. Using the right image as the reference for our disparity map, we compute the disparity on row y as

$$D_r(y, x_r) = \underset{d_{\min} \leqslant d \leqslant d_{\max}}{\operatorname{argmin}} \quad c(x_r + d, x_r). \tag{A.1}$$

A.2. The left–right heuristic [24,25]

Disparity values for corresponding pixels should be equal in absolute value but have opposite algebraic signs [25]. The search for a local minimum yields the minimum in each *column* of the epipolar cost matrix. Searching for a minimum along the *row* of the cost matrix would yield a disparity with respect to the left epipolar line. This suggests an algorithm that requires the left and right disparities to agree; that is, if the minimum in row k occurs in column j then we require that the minimum in column j be row k. One stereo algorithm [24] used

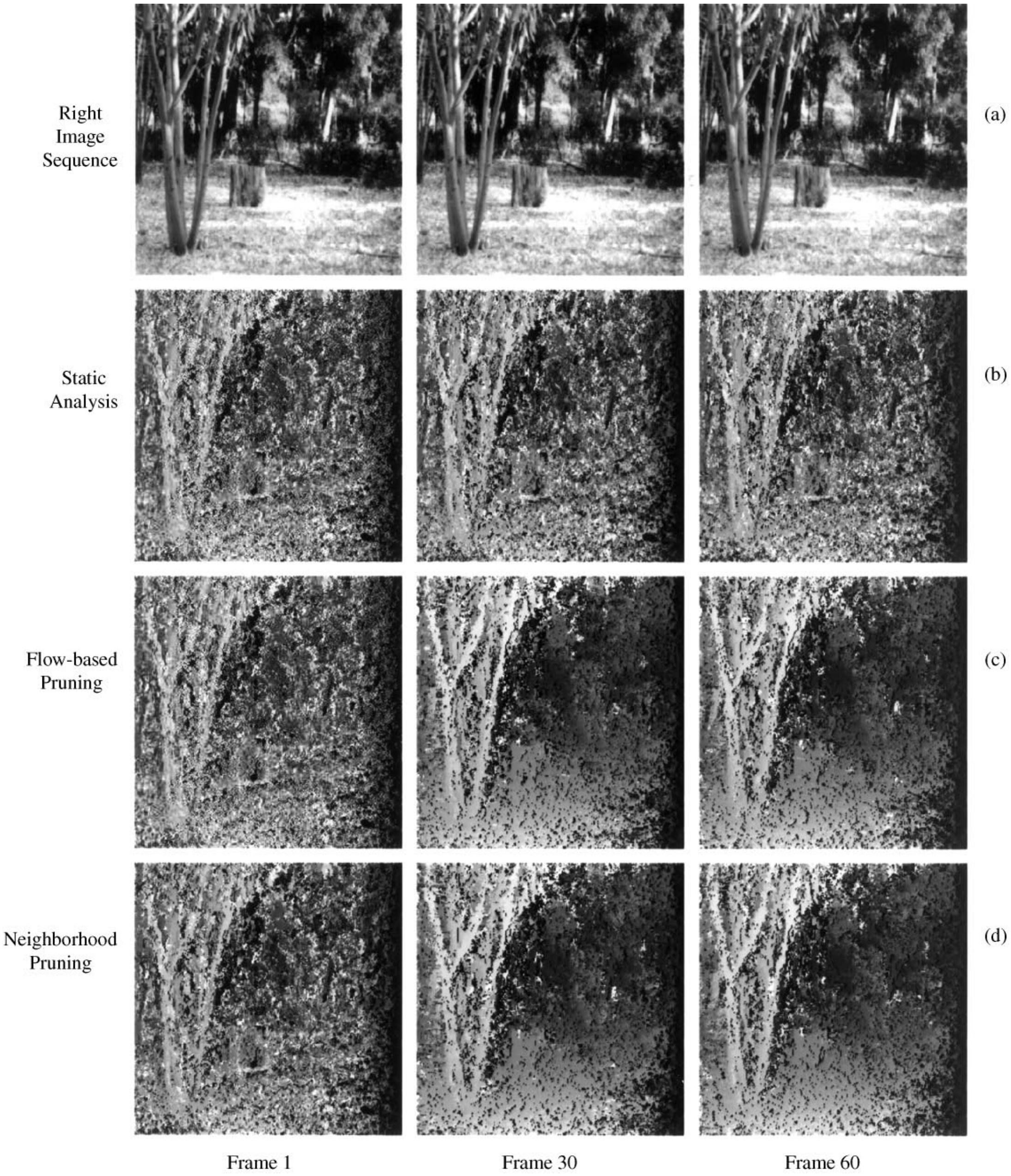


Fig. 11. A real stereo scene [256 × 256] where every pixel was corrupted by Gaussian distributed noise with zero-mean and $\sigma = 5$. The left image (not shown) is translated one pixel horizontally and one pixel vertically in every second frame and rotated by 1° in each third frame to simulate camera misalignment. In row (b) we show a static analysis. In row (c) we show the results using the optical-flow based pruning scheme. In row (d) we give the local-neighborhood pruning results. The resulting disparity map in the 60th frame is visibly superior to the static analysis. We have used 3 × 3 windows with the greedy matching algorithm, $C_O = 2.2 \times 10^4$, and a disparity range of 100 pixels.

a strategy to refine disparity estimates by horizontally flipping the left and right images and then re-analyzing them by using the flipped left image as the new right image and the flipped right image as the new left image. Instead of refining estimates, we save those matches that agree between the right and left epipolar lines. We call this method the *left–right heuristic* (LRH).[7] We define two vectors LeftMate and RightMate whose elements are the index of the matching pixel on the opposite epipolar line. If separate minimizations along rows and columns produced the same match, then we would have RightMate[LeftMate[i]] = i for each row *i*.

A.3. Dynamic programming [11,26]

Dynamic programming (DP) is a method of organizing an optimization problem to exploit the recursive structure of necessary calculations [27]. Several studies have used this matching approach for stereo matching (e.g. Refs. [11,28]). Central to the use of DP is the *monotonic ordering constraint* which states that if x_{r_i} matches x_{l_j} then $x_{r_{i+1}}$ can only match $x_{l_{j+k}}$ with $k \geqslant 1$. Without the monotonic ordering constraint, one is required to perform an exhaustive search. The computational complexity of the dynamic programming approach with the monotonic ordering constraint is $O(\delta N)$ [11].

Fig. 3 shows a graph on which the shortest path between the vertices labeled S and T can be calculated using DP. The dotted edges allow for occlusions to occur; their weights are given by the occlusion cost C_O. We associate the weight from the epipolar cost matrix, $c(x_l, x_r)$, with the diagonal (solid) edges. While a Dijkstra-type shortest path algorithm [27] solves this problem, a more efficient algorithm is possible because of the regular structure of the graph. For a detailed discussion see Refs. [11,28].

A.4. Maximum weighted matching

Fig. 2 shows a graph representation of the matching problem. With each edge we associate the weight $a(x_l, x_r) = C_M - c(x_l, x_r)$ where C_M is an upper bound on the appropriate similarity function. We have also linked an "occlusion" node to each node in right epipolar line with edge cost $C_M - C_O$. By doing this, we can include the occlusion cost as a matchable node. Alternatively, we could have linked the occlusion nodes to the left epipolar line which would result in a different matching; however, our experiments have shown that the differences in final matchings using the two linking schemes are insignificant. The same is true for adding *both* left and right occlusion nodes, with each edge having the occlusion cost $C_M - (C_O/2)$. The calculation of disparities can be viewed as a weighted matching problem on a bipartite graph. We desire to find the set of edges for which the sum of these chosen edge weights is maximum while no two edges touch the same vertex (pixel).

A.5. A greedy algorithm for weighted matchings

Greedy algorithms represent a well-known alternative to global optimization procedures [27]. This method builds a matching by progressively adding edges with maximum weights in the cost matrix $a(x_l, x_r)$ that satisfy the criteria for a feasible matching. This method differs from the maximum weighted matching approach in that the sum of the matching weights is not guaranteed to be maximum.

Given a bipartite graph $G = (V, U, E)$ and an associated cost with each edge, c_e, $e \in E$, the *greedy matching algorithm* proceeds as follows: Let $\tilde{E} = \emptyset$ and $M = E$. While there exists $e \in M, e \notin \tilde{E}$, such that $\tilde{E} \cup e$ is a feasible matching, let $e = \operatorname{argmax}\{c_e\colon \forall e \in M\}$, and set $\tilde{E} = \tilde{E} \cup e$ and $M = M - \{e\}$.

Appendix B. Occlusion cost

Based on Refs. [1,11], we present a decision theoretic estimate for an occlusion.

B.1. A decision theoretic occlusion cost

Let z_{1,i_1} and z_{2,i_2} be measurements from corresponding epipolar lines in two cameras. If the measurements are matching (i.e., they are projections of the same point/surface in space), we assume that the difference between the measurements is an independent normal random variable $z_{1,i_1} - z_{2,i_2} \sim \mathrm{N}(0, \sigma^2)$. We define $z_{i_1,i_2} = (z_{1,i_1} - z_{2,i_2})/2\sigma$ which is $\mathrm{N}(0, \frac{1}{4})$. The random variable, $z^2_{i_1,i_2}$ has a gamma distribution [29]

$$f_x(x) = \frac{c^b}{\Gamma(b)} x^{b-1} \mathrm{e}^{-cx}, \tag{B.1}$$

where $b = \frac{1}{2}$ and $c = 2$. Knowing the underlying distribution (B.1) of the matching cost, the probability P_D of obtaining a matching cost between 0 and $\tilde{C}_O$ can be calculated through the integral

$$P_D = \int_0^{\tilde{C}_O} \sqrt{\frac{2}{\pi}} x^{-1/2} \mathrm{e}^{-2x} \, \mathrm{d}x = \gamma\left(\frac{1}{2}, 2\tilde{C}_O\right), \tag{B.2}$$

where γ is the incomplete gamma function defined by

$$\gamma(b, x) = \frac{1}{\Gamma(b)} \int_0^x t^{b-1} \mathrm{e}^{-t} \, \mathrm{d}t.$$

Conversely, we may fix P_D in Eq. (B.2) and compute $\tilde{C}_O$ through tabulated percentile values of the cumulative gamma distribution or through numerical evaluation of the integral in Eq. (B.2). In this case, each matching

[7] Agreement between left and right disparities is less likely when the surface is sharply sloping away from the stereo cameras.

instance may be considered a hypothesis test where we compare the test statistic (the matching cost) to the threshold $\tilde{C}_O$ which is determined by P_D.[8]

B.2. Extending the algorithm of Ref. [11] *to accommodate windows*

Let the current epipolar line be j_0. The neighborhood $\Omega_{s,i_0,j_0}(p,q)$ in camera s, on epipolar line j_0, at the i_0th feature z_{s,i_0,j_0}, is defined as $\Omega_{s,i_0,j_0}(p,q) = [\{z_{s,i_0+i,j_0+j}\}$, s.t. $-p \leqslant i \leqslant p,\ -q \leqslant j \leqslant q]$.

For the epipolar line j_0, we compare corresponding features in the neighborhoods $\Omega_{1,i_1,j_0}(p,q)$ and $\Omega_{2,i_2,j_0}(p,q)$. The cost of matching two features is defined as the sum of squared differences over the neighborhood (i.e., a $2p+1$ by $2q+1$ rectangular window of pixels),

$$g_{j_0,i_1,i_2} = \sum_{i=-q}^{i=q} \sum_{j=-p}^{j=p} \left(\frac{z_{1,i_1+i,j_0+j} - z_{2,i_2+i,j_0+j}}{2\sigma} \right)^2. \qquad \text{(B.3)}$$

The local window has $n = (2p+1)(2q+1)$ elements and the difference between matching features is assumed to be an independent and identically distributed normal random variable, $\mathrm{N}(0, \sigma^2)$. Each term in the sum is a random variable having a gamma distribution with parameters $b = n/2$ and $c = 2$. Hence, the sum in (B.3) again has a gamma distribution (B.1) with the parameters $b = n/2$ and $c = 2$. The probability of detection, P_D and the cost of occlusion $\hat{C}_O$ are related through the incomplete gamma function

$$P_D = \gamma\left(\frac{n}{2}, 2\hat{C}_O\right). \qquad \text{(B.4)}$$

Appendix C. Bounds on the derivative of disparity

It can be shown (e.g. Refs. [1,10,30]) that for parallel stereo cameras, the following relation between range and disparity holds:

$$d = \frac{bf}{z}, \qquad \text{(C.1)}$$

where b is the baseline between cameras, f is the focal length of each camera, z is the range in world coordinates, and d is the disparity. Taking the derivative with respect to time yields

$$\dot{d} = -\frac{bf}{z^2}\dot{z}. \qquad \text{(C.2)}$$

If we assume $|\dot{z}| \leqslant Mz$ for some $M \geqslant 0$, and substitute $z = bf/d$ into Eq. (C.2), we obtain

$$|\dot{d}| \leqslant M|d|. \qquad \text{(C.3)}$$

Our assumption about $\dot{z}$ can be interpreted as a bound on the rate of approach, depending upon how far an object is located from the stereo cameras. Objects farther from the cameras can be moving toward or away from the stereo platform at a faster rate than close objects. By rewriting $M = k/d_{MAX}$ in Eq. (C.3) we obtain

$$|\dot{d}| \leqslant k\left|\frac{d}{d_{MAX}}\right|. \qquad \text{(C.4)}$$

Finally, to account for the discrete matching algorithms that we use (see Appendix A), we apply the ceiling operator to Eq. (C.4) with a small positive perturbation $\varepsilon \in (0,1)$,

$$|\dot{d}| \leqslant \left\lceil k\left|\frac{d}{d_{MAX}}\right| + \varepsilon \right\rceil, \qquad \text{(C.5)}$$

where ε prevents the case $d = 0$ from yielding a search range of zero. We set $\varepsilon = 0.5$. The parameter k can be interpreted as a bound on the maximum pixel displacement between frames. A different interpretation is possible if we assume $|\dot{z}| \leqslant M$ and substitute $z = bf/d$ into Eq. (C.2):

$$|\dot{d}| \leqslant \frac{M}{bf} d^2. \qquad \text{(C.6)}$$

Here M is a bound on the range derivative (e.g., a maximum rate of approach). Thus we have an alternative to the parameter in Eq. (5):

$$\varepsilon_{ij} = \left\lceil \tilde{k}\left(\frac{\hat{d}_{ij}}{d_{MAX}}\right)^2 + \varepsilon \right\rceil, \quad \hat{d}_{ij} \in \Omega_s. \qquad \text{(C.7)}$$

The magnitudes of both k and $\tilde{k}$ determine the maximum magnitude of the search range ε_{ij}; however, the units of k and $\tilde{k}$ differ. We have found the variations in algorithm performance are negligible when Eq. (5) is replaced by Eq. (C.7).

References

[1] G. Fielding, Matching in dynamic stereo image sequences, Ph.D. Thesis, Drexel University, 1999.

[2] S.T. Barnard, Stereo Matching, Academic Press, New York, NY, 1993, pp. 245–271 (Chapter 10).

[3] B.K.P. Horn, B.G. Schunck, Determining optical flow, Artif. Intell. 17 (1981) 185–203.

[4] A. Singh, Optic Flow Computation: A Unified Perspective, IEEE Press, Los Alamitos, CA, 1990.

[5] A. Murat Tekalp, Digital Video Processing, Prentice-Hall, Upper Saddle River, NJ, 1995.

[8] The significance of the hypothesis test would be $1 - P_D$.

[6] A.M. Waxman, J.H. Duncan, Binocular image flows: steps toward stereo-motion fusion, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 715–729.

[7] J.H. Duncan, L. Li, W.D. Wang, Recovering the three-dimensional velocity and establishing stereo correspondence from binocular image flows, Opt. Engng. 34 (7) (1995) 2157–2167.

[8] L. Matthies, R. Szeliski, T. Kanade, Kalman filter-based algorithms for estimating depth from image sequences, Int. J. Comput. Vision 3 (3) (1989) 209–238.

[9] R.Y. Tsai, A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses, IEEE J. Robotics Automat. RA-3 (4) (1987) 323–344.

[10] N. Alvertos, D. Brzakovic, R.C. Gonzalez, Camera geometries for image matching in 3-d machine vision, Trans. Pattern Anal. Mach. Intell. 11 (9) (1989) 897–915.

[11] I. Cox, S. Hingorani, S. Rao, B. Maggs, A maximum likelihood stereo algorithm, CVGIP:Image Understanding 63 (3) (1996) 542–567.

[12] G. Fielding, M. Kam, Computing the cost of occlusion, Comput. Vision Image Understanding (2000), in press.

[13] T. Kanade, Development of a video rate stereo machine, Proceedings of the 1994 ARPA Image Understanding Workshop, November 1994, pp. 549–558.

[14] H.K. Nishihara, PRISM: a practical real-time imaging stereo matcher, Technical Report 780, MIT AI Lab 1984.

[15] Q.X. Wu, A correlation-relaxation-labeling framework for computing optical flow — template matching from a new perspective, IEEE Trans. Pattern Anal. Mach. Intell. 17 (9) (1995) 843–853.

[16] P. Anandan, A computational framework and an algorithm for the measurement of visual motion, Int. J. Comput. Vision 2 (3) (1989) 283–310.

[17] Stephen T. Barnard, William B. Thompson, Disparity analysis of images, IEEE Trans. Pattern Anal. Mach. Intell. 2 (4) (1980) 333–340.

[18] G. Pajares, J.M. Cruz, J. Aranda, Relaxation by Hopfield network in stereo image matching, Pattern Recognition 31 (5) (1998) 561–574.

[19] T.A. Poggio, V. Torre, C. Koch, Computational vision and regularization theory, Nature 317 (1985) 314–319.

[20] K. Prazdny, Detection of binocular disparities, Biol. Cybernet. 52 (1985) 93–99.

[21] T. Kanade, M. Okutomi, A stereo matching algorithm with an adaptive window: theory and experiment, IEEE Trans. Pattern Anal. Mach. Intell. 16 (9) (1994) 920–932.

[22] R. Sarpeshkar, J. Kramer, G. Indiveri, C. Koch, Analog VLSI architectures for motion processing: From fundamental limits to system applications, Proc. IEEE 84 (7) (1996) 969–987.

[23] B. Ross, A practical stereo vision system, Proceedings of the IEEE Computer Vision and Pattern Recognition, 1993, pp. 148–153.

[24] M.J. Hannah, Sri's baseline stereo system, Proceedings of the 1985 DARPA, Image Understanding Workshop, 1985, pp. 149–155.

[25] D.G. Jones, J. Malik, Computational framework for determining stereo correspondence from a set of linear spatial filters, European Conference on Computer Vision, 1992, pp. 395–410.

[26] H.H. Baker, T.O. Binford, Depth from edge and intensity based stereo, International Joint Conference on Artificial Intelligence, 1981, pp. 631–636.

[27] T. Cormen, C. Leiserson, R. Rivest, Introduction to Algorithms, MIT Press, Cambridge, MA, 1990.

[28] Y. Ohta, T. Kanade, Stereo by intra- and inter-scanline search using dynamic programming, IEEE Trans. Pattern Anal. Mach. Intell. 7 (2) (1985) 139–154.

[29] A. Papoulis, Probability, Random Variables, and Stochastic Processes, 3rd Edition, McGraw-Hill, New York, NY, 1991.

[30] U.R. Dhond, J.K. Aggarwal, Structure from stereo: a review, IEEE Transaction Systems Man Cybernet. 19 (6) (1989) 1489–1510.

About the Author—GABRIEL FIELDING received the M.S. (Electrical Engineering) and Ph.D. degrees from Drexel University, in 1996 and 1999 respectively. From 1995 through 1998, he was a National Science and Defense Graduate Research Fellow sponsored by the Office of Naval Research. He has worked on computer vision projects with the Advanced Robotics Group at the Naval Research Labs in San Diego, CA. He is currently a research scientist with the Eastman Kodak Company in Rochester, NY. His research interests are motion estimation and parallel algorithms for image processing.

About the Author—MOSHE KAM was educated at Tel Aviv University (B.S 1977) and Drexel University (M.Sc. 1985, Ph.D. 1987). At present he is a Professor of Electrical and Computer Engineering at Drexel, and Director of its Data Fusion Laboratory.

He is a recipient of an NSF Presidential Young Investigator Award (1990), the C.H. MacDonald award for the Outstanding Young Electrical Engineering Educator (1991), and the Drexel University Research Award (1998). His research interests are in System Theory, Detection and Estimation (especially distributed detection and decision fusion) Robotics, Navigation, and Control.