International Journal of Computer Vision, 3, 209–236 (1989)

Kalman Filter-based Algorithms for Estimating Depth from Image Sequences

LARRY MATTHIES AND TAKEO KANADE
Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213; Schlumberger Palo Alto Research, 3340 Hillview Ave., Palo Alto, CA 94304

RICHARD SZELISKI
Digital Equipment Corporation, 1 Kendall Square, Building 700, Cambridge, MA 02139

Abstract
Using known camera motion to estimate depth from image sequences is an important problem in robot vision. Many applications of depth-from-motion, including navigation and manipulation, require algorithms that can estimate depth in an on-line, incremental fashion. This requires a representation that records the uncertainty in depth estimates and a mechanism that integrates new measurements with existing depth estimates to reduce the uncertainty over time. Kalman filtering provides this mechanism. Previous applications of Kalman filtering to depth-from-motion have been limited to estimating depth at the location of a sparse set of features. In this paper, we introduce a new, pixel-based (*iconic*) algorithm that estimates depth and depth uncertainty at each pixel and incrementally refines these estimates over time. We describe the algorithm and contrast its formulation and performance to that of a feature-based Kalman filtering algorithm. We compare the performance of the two approaches by analyzing their theoretical convergence rates, by conducting quantitative experiments with images of a flat poster, and by conducting qualitative experiments with images of a realistic outdoor-scene model. The results show that the new method is an effective way to extract depth from lateral camera translations. This approach can be extended to incorporate general motion and to integrate other sources of information, such as stereo. The algorithms we have developed, which combine Kalman filtering with iconic descriptions of depth, therefore can serve as a useful and general framework for low-level dynamic vision.

1 Introduction

Using known camera motion to estimate depth from image sequences is important in many applications of computer vision to robot navigation and manipulation. In these applications, depth-from-motion can be used by itself, as part of a multimodal sensing strategy, or as a way to guide stereo matching. Many applications require a depth estimation algorithm that operates in an on-line, incremental fashion. To develop such an algorithm, we require a depth representation that includes not only the current depth estimate, but also an estimate of the uncertainty in the current depth estimate.

Previous work [3, 5, 9, 10, 16, 17, 25] has identified Kalman filtering as a viable framework for this problem, because it incorporates representations of uncertainty and provides a mechanism for incrementally reducing uncertainty over time. To date, applications of this framework have largely been restricted to estimating the positions of a sparse set of trackable features, such as points or line segments. While this is adequate for many robotics applications, it requires reliable feature extraction and it fails to describe large areas of the image. Another line of work has addressed the problem of extracting dense displacement or depth estimates from image sequences. However, these previous approaches have either been restricted to two-frame analysis [1] or have used batch processing of the image sequence, for example via spatiotemporal filtering [11].

In this paper we introduce a new, pixed-based (*iconic*) approach to incremental depth estimation and compare it mathematically and experimentally to a feature-based approach we developed previously [16]. The new approach represents depth and depth variance at every pixel and uses Kalman filtering to extrapolate and update the pixel-based depth representation. The algorithm uses correlation to measure the optical flow and to estimate the variance in the flow, then uses the known camera motion to convert the flow field into a depth map. It then uses the Kalman filter to generate an updated depth map from a weighted combination of the new measurements and the prior depth estimates. Regularization is employed to smooth the depth map

and to fill in the underconstrained areas. The resulting algorithm is parallel, uniform, and can take advantage of mesh-connected or multiresolution (pyramidal) processing architectures.

The remainder of this paper is structured as follows. In the next section, we give a brief review of Kalman filtering and introduce our overall approach to Kalman filtering of depth. Next, we review the equations of motion, present a simple camera model, and examine the potential accuracy of the method by analyzing its sensitivity to the direction of camera motion. We then describe our new, pixel-based depth-from-motion algorithm and review the formulation of the feature-based algorithm. Next, we analyze the theoretical accuracy of both methods, compare them both to the theoretical accuracy of stereo matching, and verify this analysis experimentally using images of a flat scene. We then show the performance of both methods on images of realistic outdoor scene models. In the final section, we discuss the promise and the problems involved in extending the method to arbitrary motion. We also conclude that the ideas and results presented apply directly to the much broader problem of integrating depth information from multiple sources.

2 Estimation Framework

The depth-from-motion algorithms described in this paper use image sequences with small frame-to-frame camera motion [4]. Small motion minimizes the correspondence problem between successive images, but sacrifices depth resolution because of the small baseline between consecutive image pairs. This problem can be overcome by integrating information over the course of the image sequence. For many applications, it is desirable to process the images incrementally by generating updated depth estimates after each new image is acquired, instead of processing many images together in a batch. The incremental approach offers real-time operation and requires less storage, since only the current estimates of depth and depth uncertainty need to be stored.

The Kalman filter is a powerful technique for doing incremental, real-time estimation in dynamic systems. It allows for the integration of information over time and is robust with respect to both system and sensor noise. In this section, we first present the notation and the equations of the Kalman filter, along with a simple example. We then sketch the application of this framework to motion-sequence processing and discuss those parts of the framework that are common to both the iconic and the feature-based algorithms. The details of these algorithms are given in sections 4 and 5, respectively.

2.1. Kalman Filter

The Kalman filter is a Bayesian estimation technique used to track stochastic dynamic systems being observed with noisy sensors. The filter is based on three separate probabilistic models, as shown in table 1. The first model, the *system model,* describes the evolution over time of the current state vector u_t. The transition between states is characterized by the known transition matrix Φ_t and the addition of Gaussian noise with a covariance Q_t. The second model, the *measurement* (or *sensor*) *model,* relates the measurement vector d_t to the current state through a measurement matrix H_t and the addition of Gaussian noise with a covariance R_t. The third model, the *prior model,* describes the knowledge about the system state $\hat{u}_0$ and its covariance P_0 before the first measurement is taken. The sensor and process noise are assumed to be uncorrelated.

Table 1. Kalman filter equations.

Models	system model	$u_t = \Phi_{t-1}u_{t-1} + \eta_t,\ \eta_t \sim N(0, Q_t)$
	measurement model	$d_t = H_t u_t + \xi_t, \xi_t \sim N(0, R_t)$
	prior model	$E[u_0] = \hat{u}_0,\ \mathrm{cov}[u_0] = P_0$
	(other assumptions)	$E[\eta_i \xi_j^T] = 0$
Prediction phase	state estimate extrapolation	$\hat{u}_t^- = \Phi_{t-1}\hat{u}_{t-1}^+$
	state covariance extrapolation	$P_t^- = \Phi_{t-1}P_{t-1}^+\Phi_{t-1}^T + Q_{t-1}$
Update phase	state estimate update	$\hat{u}_t^+ = \hat{u}_t^- + K_t[d_t - H_t\hat{u}_t^-]$
	state coviarance update	$P_t^+ = [I - K_tH_t]P_t^-$
	Kalman gain matrix	$K_t = P_t^-H_t^T[H_tP_t^-H_t^T R_t]^{-1}$

To illustrate the equations of table 1, we will use the example of a ping-pong-playing robot that tracks a moving ball. In this example, the state consists of the ball position and velocity, $u = [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}\ 1]^T$, where x and y lie parallel to the image plane (y is up), and z is parallel to the optical axis. The state transition matrix models the ball dynamics, for example by the matrix

$$\Phi_t = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & -\beta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\beta & 0 & -g\Delta t \\ 0 & 0 & 0 & 0 & 0 & -\beta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where Δt is the time step, β is the coefficient of friction and g is gravitational acceleration. The process noise matrix Q_t models the random disturbances that influence the trajectory. If we assume that the camera uses orthographic projection and uses a simple algorithm to find the "center of mass" (x,y) of the ball, then the sensor can then be modeled by the matrix

$$H_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

which maps the state u to the measurement d. The uncertainty in the sensed ball position can be modeled by a 2×2 covariance matrix R_t.

Once the system, measurement, and prior models have been specified (i.e., the upper third of table 1), the Kalman filter algorithm follows from the formulation in the lower two thirds of table 1. The algorithm operates in two phases: extrapolation (prediction) and update (correction). At time t, the previous state and covariance estimates, $\hat{u}_{t-1}^+$ and P_{t-1}^+, are extrapolated to predict the current state $\hat{u}_t^-$ and covariance P_t^-. The predicted covariance is used to compute the new Kalman gain matrix K_t and the updated covariance matrix P_t^+. Finally, the measurement residual $d_t - H_t\hat{u}_t^-$ is weighted by the gain matrix K_t and added to the predicted state u_t^- to yield the updated state u_t^+. A block diagram for the Kalman filter is given in figure 1.

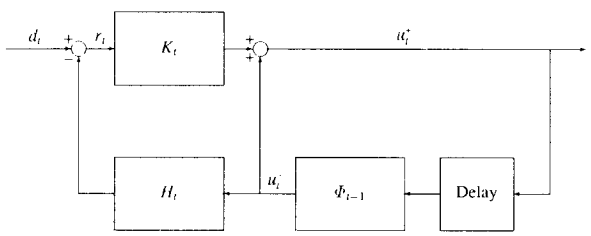


Fig. 1. Kalman filter block diagram.

2.2. Application to Depth from Motion

To apply the Kalman filter estimation framework to the depth-from-motion problem, we specialize each of the three models (system, measurement, and prior) and define the implementations of the extrapolation and update stages. This section briefly previews how these components are chosen for the two depth-from-motion algorithms described in this paper. The details of the implementation are left to sections 4 and 5.

The first step in designing a Kalman filter is to specify the elements of the state vector. The iconic depth-from-motion algorithm estimates the depth at each pixel in the current image, so the state vector in this case is the entire depth map.[1] Thus, the diagonal elements of the state covariance matrix P_t are the variances of the depth estimates at each pixel. As discussed shortly, we implicitly use off-diagonal elements of the inverse covariance matrix P_0^{-1} as part of the update stage of the filter, but do not explicitly model them anywhere in the algorithm because of the large size of the matrix. For the feature-based approach, which tracks edge elements through the image sequence, the state consists of a 3D position vector for each feature. We model the full covariance matrix of each individual feature, but treat separate features as independent.

The system model in both approaches is based on the same motion equations (section 3.1), but the implementations of the extrapolation and update stages differ because of the differences in the underlying representations. For the iconic method, the extrapolation stage uses the depth map estimated for the current frame, together with knowledge of the camera motion, to predict the depth and depth variance at each pixel in the next frame. Similarly, the update stage uses measurements of depth at each pixel to update the depth and variance estimates at each pixel. For the feature-based method, the extrapolation stage predicts the position vector and covariance matrix of each feature for the next image, then uses measurements of the image coordinates of the feature to update the position vector and the covariance matrix. Details of the measurement models for each algorithm will be discussed later.

Finally, the prior model can be used to embed prior knowledge about the scene. For the iconic method, for example, smoothness constraints requiring nearby image points to have similar disparity can be modeled easily by off-diagonal elements of the inverse of the prior covariance matrix P_0 [29]. Our algorithm incorporates

[1]Our actual implementation uses inverse depth (called "disparity") See section 4.

this knowledge as part of a smoothing operation that follows the state update stage. Similar concepts may be applicable to modeling figural continuity [20,24] in the edge-tracking approach, that is, the constraint that connected edges must match connected edges; however, we have not pursued this possibility.

3 Motion Equations and Camera Model

Our system and measurement models are based on the equations relating scene depth and camera motion to the induced image flow. In this section, we review these equations for an idealized camera (focal length = 1) and show how to use a simple calibration model to relate the idealized equations to real cameras. We also derive an expression for the relative uncertainty in depth estimates obtained from lateral versus forward camera translation. This expression shows concretely the effects of camera motion on depth uncertainty and reinforces the need for modeling the uncertainty in computed depth.

3.1. Equations of Motion

If the inter-frame camera motion is sufficiently small, the resulting optical flow can be expressed to a good approximation in terms of the instantaneous camera velocity [6, 13, 33]. We will specify this in terms of a translational velocity **T** and an angular velocity **R**. In the camera coordinate frame (figure 2), the motion of a 3D point **P** is described by the equation

$$\frac{d\mathbf{P}}{dt} = -\mathbf{T} - \mathbf{R} \times \mathbf{P}$$

Expanding this into components yields

$$\begin{aligned} dX/dt &= -T_x - R_y Z + R_z Y \\ dY/dt &= -T_y - R_z X + R_x Z \\ dZ/dt &= -T_z - R_x Y + R_y X \end{aligned} \qquad [1]$$

Now, projecting (X, Y, Z) onto an ideal, unit focal length image,

$$x = \frac{X}{Z} \qquad y = \frac{Y}{Z} \; .$$

taking the derivatives of (x, y) with respect to time, and substituting in from equation (1) leads to the familiar equations of optical flow [33]:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix} \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} \qquad [2]$$

These equations relate the depth Z of the point to the camera motion $\mathbf{T}, \mathbf{R}$ and the induced image displacements or optical flow $[\Delta x \; \Delta y]^T$. We will use these equations to measure depth, given the camera motion and optical flow, and to predict the change in the depth map between frames. Note that parameterizing (2) in terms of the inverse depth $d = 1/Z$ makes the equations linear in the "depth" variable. Since this leads to a simpler estimation formulation, we will use this parameterization in the balance of the paper.

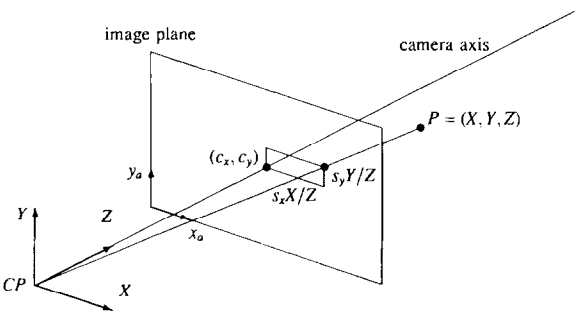


Fig. 2. Camera model. *CP* is the center of projection.

3.2. Camera Model

Relating the ideal flow equations to real measurements requires a camera model. If optical distortions are not severe, a pin-hole camera model will suffice. In this paper we adopt a model similar to that originated by Sobel [27] (figure 2). This model specifies the origin (c_x, c_y) of the image coordinate system and a pair of scale factors (s_x, s_y) that combine the focal length and image aspect ratio. Denoting the actual image coordinates with a subscript a, the projection onto the actual image is summarized by the equation

$$\begin{bmatrix} x_a \\ y_a \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} s_x & 0 & c_x \\ 0 & s_y & c_y \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{Z} \, CP \qquad [3]$$

C is known as the *collimation matrix*. Thus, the ideal image coordinates (x, y) are related to the actual image coordinates by

$$x_a = s_x x + c_x \qquad y_a = s_y y + c_y$$

Equations in the balance of the paper will primarily use ideal image coordinates for clarity. These equations can be re-expressed in terms of actual coordinates using the transformations above.

3.3. Sensitivity Analysis

Before describing our Kalman filter algorithms, we will analyze the effect of different camera motions on the uncertainty in depth estimates. Given specific descriptions of real cameras and scenes, we can obtain bounds on the estimation accuracy of depth-from-motion algorithms using perturbation or covariance analysis techniques based on first-order Taylor expansions [8]. For example, if we solve the motion equations for the inverse depth d in terms of the optical flow, camera motion, and camera model,

$$d = F(\Delta x, \Delta y, T, R, c_x, c_y, s_x, s_y) \qquad [4]$$

then the uncertainty in depth arising from uncertainty in flow, motion, and calibration can be expressed by

$$\delta d = J_f \delta f + J_m \delta m + J_c \delta c \qquad [5]$$

where J_f, J_m, and J_c are the Jacobians of (4) with respect to the flow, motion, and calibration parameters, respectively, and δf, δm, and δc are perturbations of the respective parameters. We will use this methodology to draw some conclusions about the relative accuracy of depth estimates obtained from different classes of motion.

It is well known that camera rotation provides no depth information. Furthermore, for a translating camera, the accuracy of depth estimates increases with increasing distance of image features from the *focus of expansion* (FOE), the point in the image where the translation vector (**T**) pierces the image. This implies that the 'best' translations are parallel to the image plane and that the 'worst' are forward along the camera axis. We will give a short derivation that demonstrates the relative accuracy obtainable from forward and lateral camera translation. The effects of measurement uncertainty on depth-from-motion calculations is also examined in [26].

For clarity, we consider only one-dimensional flow induced by translation along the X or Z axes. For an ideal camera, lateral motion induces the flow

$$\Delta x_l = \frac{-T_x}{Z} \qquad [6]$$

whereas forward motion induces the flow

$$\Delta x_f = \frac{xT_z}{Z} \qquad [7]$$

The inverse depth (or disparity) in each case is

$$d_l = \frac{1}{Z} = \frac{-\Delta x_l}{T_x}$$

$$d_f = \frac{\Delta x_f}{xT_z}$$

Therefore, perturbations of δx_l and δx_f in the flow measurements Δx_l and Δx_f yield the following perturbations in the disparity estimates:

$$\delta d_l = \frac{\delta x_l}{|T_x|}$$

$$\delta d_f = \frac{\delta x_f}{|xT_z|}$$

These equations give the error in the inverse depth as a function of the error in the measured image displacement, the amount of camera motion, and position of the feature in the field of view. Since we are interested in comparing forward and lateral motions, a good way to visualize these equations is to plot the relative depth uncertainty, $\delta d_f / \delta d_l$. Assuming that the flow perturbations δx_l and δx_f are equal, the relative uncertainty is

$$\frac{\delta d_f}{\delta d_l} = \frac{\delta x_f / |xT_z|}{\delta x_l / |T_x|} = \frac{|T_x|}{|xT_z|}$$

The image coordinate x indicates where the object appears in the field of view. Figure 3 shows that x equals the tangent of the angle θ between the object and the camera axis. The formula for the relative uncertainty is thus

$$\frac{\delta d_f}{\delta d_l} = \frac{|T_x|}{|T_z \tan\theta|} \qquad [9]$$

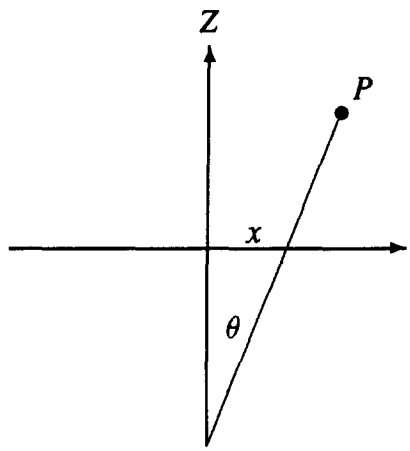


Fig. 3. Angle between object and camera axis is θ.

This relationship is plotted in figure 4 for $T_x = T_z$. At 45° from the camera axis, depth uncertainty is equal for forward and lateral translations. As this angle approaches zero, the ratio of uncertainty grows, first slowly then increasingly rapidly. As a concrete example, for the experiments in section 6.2 the field of view was approximately 36°, so the edges of the images were 18° from the camera axis. At this angle, the ratio of uncertainties is 3.1; halfway from the center to the edge of the image, at 9°, the ratio is 6.3. In general, for practical fields of view, the accuracy of depth extracted from forward motion will be effectively unusable for a large part of the image.

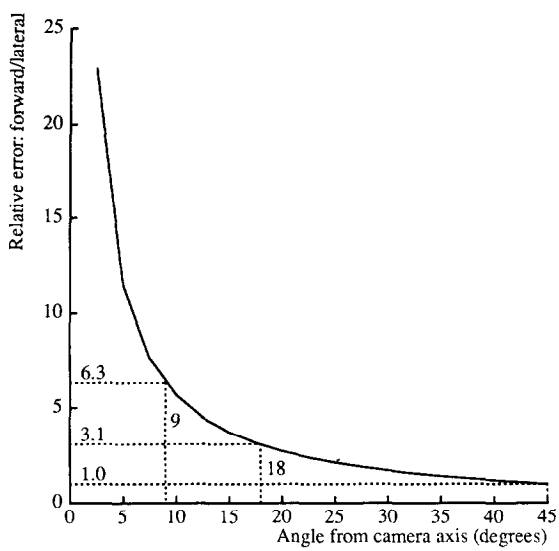


Fig. 4. Relative depth uncertainty for forward vs. lateral translation.

By setting $\delta d_f/\delta d_l = 1$, equation (9) also expresses the relative distances the camera must move forward and laterally to obtain equally precise depth estimates. An alternate interpretation for figure 4 is that it expresses the relative precision of stereo and depth-from-motion in a dynamic, binocular stereo system.

We draw several conclusions from this analysis. First, it underscores the value of representing depth uncertainty as we describe in the following sections. Second, for practical depth estimation, forward motion is effectively unusable compared with lateral motion. Finally, we can relate these results to dynamic, binocular stereo by noting that depth from forward motion will be relatively ineffective for constraining or confirming binocular correspondence.

4 Iconic Depth Estimation

This section describes the incremental, iconic depth-estimation algorithm we have developed. The algorithm processes each new image as it arrives, extracting optical flow at each pixel using the current and previous intensity images, then integrates this new information with the existing depth estimates.

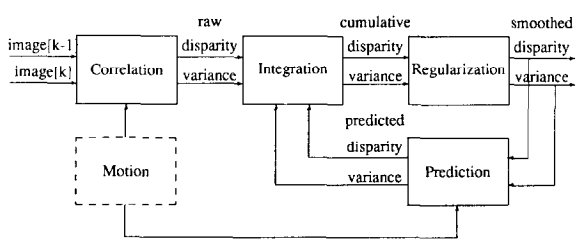


Fig. 5. Iconic depth-estimation block diagram.

The algorithm consists of four main stages (figure 5). The first stage uses correlation to compute estimates of optical flow vectors and their associated covariance matrixes. These are converted into disparity (inverse-depth) measurements using the known camera motion. The second stage integrates this information with the disparity map predicted from the previous time step. The third stage uses regularization-based smoothing to reduce measurement noise and to fill in areas of unknown disparity. The last stage uses the known camera motion to predict the disparity field that will be seen in the next frame and resamples the field to keep it iconic (pixel-based).

4.1. Measuring Disparity

The first stage of the Kalman filter computes measurements of disparity from the difference in intensity between the current image and the previous image. This computation proceeds in two parts. First, a two-dimensional optical flow vector is computed at each point using a correlation-based algorithm. The uncertainty in this vector is characterized by a bivariate Gaussian distribution. Second, these vectors are converted into disparity measurements using the known camera motion and the motion equations developed in section 3.1.

This two-part formulation is desirable for several reasons. First, it allows probabilistic characterizations of uncertainty in flow to be translated into probabilistic characterizations of uncertainty in disparity. This is especially valuable if the camera motion is also uncertain, since the equations relating flow to disparity can be extended to model this as well [25]. Second, by characterizing the level of uncertainty in the flow, it allows us to evaluate the potential accuracy of the algorithm independent of how flow is obtained. Finally, bivariate Gaussian distributions can capture the distinctions between knowing zero, one, or both components of flow [1, 11, 22], and therefore subsume the notion of the aperture problem.

The problem of optical flow estimation has been studied extensively. Early approaches used the ratio of the spatial and temporal image derivatives [12], while more recent approaches have used correlation between images [1] or spatiotemporal filtering [11]. In this paper we use a simple version of correlation-based matching. This technique, which has been called the *sum of squared differences* (SSD) method [1], integrates the squared intensity difference between two shifted images over a small area to obtain an error measure

$$e_t(\Delta x, \Delta y; x, y) = \iint [f_t(x - \Delta x + \lambda, y - \Delta y + \eta) - f_{t-1}(x + \lambda, y + \eta)]^2 \, d\lambda \, d\eta$$

where f_t and f_{t-1} are the two intensity images, and $w(\lambda, \eta)$ is a weighting function. The SSD measure is computed at each pixel for a number of possible flow values. In [1], a coarse-to-fine technique is used to limit the range of possible flow values. In our images the possible range of values is small (since we are using small-motion sequences), so a single-resolution algorithm suffices.[2] The resulting error surface $e_t(\Delta x, \Delta y; x, y)$ is approximately parabolic in shape. The lowest point of this surface defines the flow measurement and the shape of the surface defines the covariance matrix of the measurement.

To convert the displacement vector $[\Delta x \; \Delta y]^T$ into a disparity measurement, we assume that the camera motion $(\mathbf{T}, \mathbf{R})$ is given. The optical flow equation (2) can then be used to estimate depth as follows. First we abbreviate (2) to

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = d \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \begin{bmatrix} r_x \\ r_y \end{bmatrix} + \xi \qquad [10]$$

where d is the inverse depth and ξ is an error vector representing noise in the flow measurement. The noise ξ is assumed to be a bivariate Gaussian random vector with a zero mean and a covariance matrix P_m computed by the flow estimation algorithm. Equation (10) can be re-expressed in the following standard form for linear estimation problems:

$$\Delta \mathbf{x} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} - \begin{bmatrix} r_x \\ r_y \end{bmatrix} = d \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \xi \qquad [11]$$
$$= \mathbf{H}d + \xi$$

The optimal estimate of the disparity d is then [19]

$$d = (H^T P_m^{-1} H)^{-1} H^T P_m^{-1} \Delta \mathbf{x} \qquad [12]$$

and the variance of this disparity measurement is

$$\sigma_d^2 = (H^T P_m^{-1} H)^{-1} \qquad [13]$$

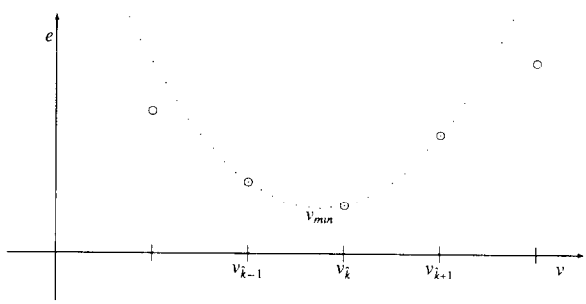


Fig. 6. Parabolic to fit to SSD error surface.

[2]It may be necessary to use a larger search range at first, but once the estimator has "latched on" to a good disparity map, the predicted disparity and disparity variance can be used to limit the search by computing confidence intervals.

This measurement process has been implemented in a simplified form, under the assumption that the flow is parallel to the image raster. To improve precision, each scan line of two successive images is magnified by a factor of four by cubic interpolation. The SSD measure e_k is computed at each interpolated subpixel displacement v_k, using a 5×5-pixel window. The minimum error $(v_{\hat{k}}, e_{\hat{k}})$ is found and a parabola

$$e(v) = av^2 + bv + c$$

is fit to this point and its two neighbors $(v_{\hat{k}-1}, e_{\hat{k}-1})$ and $(v_{\hat{k}+1}, e_{\hat{k}+1})$ (figure 6). The minimum of this parabola establishes the flow estimate (to sub-sub-pixel precision). Appendix A shows that the variance of the flow measurement is

$$\text{var}(e) = \frac{2\sigma_n^2}{a}$$

where σ_n^2 is the variance of the image noise process. The appendix also shows that adjacent flow estimates are correlated over both space and time. The significance of this fact is considered in the following two sections and in section 6.1.

4.2. Updating the Disparity Map

The next stage in the iconic depth estimator is the integration of the new disparity measurements with the predicted disparity map (this step is omitted for the first pair of images). If each value in the measured and the predicted disparity maps is not correlated with its neighbors, then the map updating can be done at each pixel independently. In this case, the covariance matrices R_t and P_t^- of table 1 are diagonal, so the matrix equations of the update phase decompose into separate scalar equations for each pixel. We will describe the procedure for this case first, then consider the consequences of correlation.

To update a pixel value, we first compute the variance of the updated disparity estimate

$$p_t^+ = [(p_t^-)^{-1} + (\sigma_d^2)^{-1}]^{-1} = \frac{p_t^- \sigma_d^2}{p_t^- + \sigma_d^2}$$

and the Kalman filter gain K

$$K = \frac{p_t^+}{\sigma_d^2} = \frac{p_t^-}{p_t^- + \sigma_m^2}$$

We then update the disparity value by using the Kalman filter update equation

$$u_t^+ = u_t^- + K(d - u_t^-)$$

where u_t^- and u_t^+ are the predicted and updated disparity estimates and d is the new disparity measurement. This update equation can also be written as

$$u_t^+ = p_t^+ \left(\frac{u_t^-}{p_t^-} + \frac{d}{\sigma_d^2} \right)$$

This shows that the updated disparity estimate is a linear combination of the predicted and measured values, inversely weighted by their respective variances.

As noted in the previous section, the depth measurements d are actually correlated over both space and time. This induces correlations in the updated depth estimates u_t^+ and implies that the measurement covariance matrix R_t and the updated state covariance matrix P_t^+ will not be diagonal in a complete stochastic model for this problem. We currently do not model these correlations because of the large expense involved in computing and storing the entire covariance matrices. Finding more concise models of the correlation is a subject for future research.

4.3. Smoothing the Map

The raw depth or disparity values obtained from optical flow measurements can be very noisy, especially in areas of uniform intensity. We employ smoothness constraints to reduce the noise and to "fill in" underconstrained areas. The earliest example of this approach is that of Horn and Schunck [12]. They smoothed the optical flow field (u, v) by jointly minimizing the error in the flow equation

$$\varepsilon_b = E_x u + E_y v + E_t$$

(E is image intensity) and the departure from smoothness

$$\varepsilon_c^2 = |\nabla u|^2 + |\nabla v|^2$$

The smoothed flow was that which minimized the total error

$$\varepsilon^2 = \int\int (\varepsilon_b^2 + \alpha^2 \varepsilon_c^2)\, dx\, dy$$

where α is a blending constant. More recently, this approach has been formalized using the theory of regularization [31] and extended to use two-dimensional confidence measures equivalent to local covariance estimates [1, 22].

For our application, smoothing is done on the disparity field, using the inverse variance of the disparity

estimate as the confidence in each measurement. The smoother we use is the generalized piecewise continuous spline under tension [32], which uses finite element relaxation to compute the smoothed field. The algorithm is implemented with a three-level coarse-to-fine strategy to speed convergence and is amenable to implementation on a parallel computer.

Surface smoothness assumptions are violated where discontinuities exist in the true depth function, in particular at object boundaries. To reduce blurring of the depth map across such boundaries, we incorporate a discontinuity detection procedure in the smoother. After several iterations of smoothing have been performed, depth discontinuities are detected by thresholding the angle between the view vector and the local surface normal (appendix B) and doing nonmaximum suppression. This is superior to applying edge detection directly to the disparity image, because it properly takes into account the 3D geometry and perspective projection. Once discontinuities have been detected, they are incorporated into the piecewise continuous smoothing algorithm and a few more smoothing iterations are performed. Our approach to discontinuity detection, which interleaves smoothing and boundary detection, is similar to Terzopoulos' continuation method [32]. The alternative of trying to estimate the boundaries in conjunction with the smoothing [14] has not been tried, but could be implemented within our framework. An interesting issue we have not explored is the propagation of detected discontinuities between frames.

The smoothing stage can be viewed as the part of the Kalman filtering algorithm that incorporates prior knowledge about the smoothness of the disparity map. As shown in [29], a regularization-based smoother is equivalent to a prior model with a correlation function defined by the degree of the stabilizing spline (e.g., membrane or thin plate). In terms of table 1, this means that the prior covariance matrix P_0 is nondiagonal. The resulting posterior covariance matrix of the disparity map contains off-diagonal elements modeling the covariance of neighboring pixels. Note that this reflects the surface smoothness model and is distinct from the measurement-induced correlation discussed in the previous section. An optimal implementation of the Kalman filter would require transforming the prior model covariance during the prediction stage and would significantly complicate the algorithm. Our choice to explicitly model only the variance at each pixel, with covariance information implicitly modeled in a fixed regularization stage, has worked well in practice.

4.4. Predicting the Next Disparity Map

The extrapolation stage of the Kalman filter must predict both the depth and the depth uncertainty for each pixel in the next image. We will describe the disparity extrapolation first, then consider the uncertainty extrapolation.

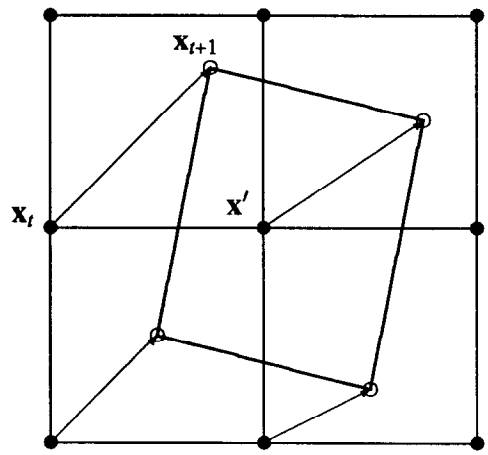


Fig. 7. Illustration of disparity prediction stage.

Our approach is illustrated in figure 7. At time t, the current disparity map and motion estimate are used to predict the optical flow between images t and $t + 1$, which in turn indicates where the pixels in frame t will 'move to' in the next frame:

$$x_{t+1} = x_t + \Delta x_t$$
$$y_{t+1} = y_t + \Delta y_t$$

The flow estimates are computed with equation (2), assuming that Z, $\mathbf{T}$, and $\mathbf{R}$ are known.[3] Next we predict what the new depth of this point will be using the equations of motion. From (2) we have

$$\begin{aligned}\Delta Z_t &= -T_z - R_x Y_t + R_y X_t \\ &= -T_z - R_x y_t Z_t + R_y x_t Z_t\end{aligned}$$

so that the predicted depth at x_{t+1}, y_{t+1} is

$$\begin{aligned}Z_{t+1} &= Z_t + \Delta Z_t \\ &= (1 - R_x y_t + R_y x_t) Z_t - T_z \\ &= \alpha Z_t - T_z\end{aligned}$$

An estimate of the inverse depth can be obtained by inverting this equation, yielding

$$u_{t+1}^- = \frac{u_t^+}{\alpha - T_z u_t^+} \qquad [14]$$

[3]There will be uncertainty in x_{t+1} and y_{t+1} due to uncertainty in the motion and disparity estimates. We ignore this for now.

This equation is nonlinear in the state variable, so it deviates from the form of linear system model illustrated in table 1. Nonlinear models are discussed in a number of references, such as [19].

In general, this prediction process will yield estimates of disparity between pixels in the new image (figure 7), so we need to resample to obtain predicted disparity at pixel locations. For a given pixel $\mathbf{x}'$ in the new image, we find the square of extrapolated pixels that overlap $\mathbf{x}'$ and compute the disparity at $\mathbf{x}'$ by bilinear interpolation of the extrapolated disparities. Note that it may be possible to detect occlusions by recording where the extrapolated squares turn away from the camera. Detecting "disocclusions," where newly visible areas become exposed, is not possible if the disparity field is assumed to be continuous, but is possible if disparity discontinuities have been detected.

Uncertainty will increase in the prediction phase due to errors from many sources, including uncertainty in the motion parameters, errors in calibration, and inaccurate models of the camera optics. A simple approach to modeling these errors is to lump them together by inflating the current variance estimates by a small multiplicative factor in the prediction stage. Thus, the variance prediction associated with the disparity prediction of equation (14) is

$$p_{t+1}^{-} = (1 + \epsilon)p_t^{+} \qquad [15]$$

In the Kalman filtering literature this is known as exponential age-weighting of measurements [19], because it decreases the weight given to previous measurements by an exponential function of time. This is the approach used in our implementation. We first inflate the variance in the current disparity map using equation (15), then warp and interpolate the variance map in the same way as the disparity map. A more exact approach is to attempt to model the individual sources of error and to propagate their effects through the prediction equations. Appendix C examines this for uncertain camera motion.

5 Feature-Based Depth Estimation

The dense, iconic depth-estimation algorithm described in the previous section can be compared with existing depth-estimation methods based on sparse feature tracking. Such methods [2, 5, 10, 16] typically define the state vector to be the parameters of the 3D object being tracked, which is usually a point or straight-line segment. The 3D motion of the object between frames defines the system model of the filter and the perspective projection of the object onto each image defines the measurement model. This implies that the measurement equations (the perspective projection) are nonlinear functions of the state variables (e.g., the 3D position vector); this requires linearization in the update equations and implies that the error distribution of the 3D coordinates will *not* be Gaussian. In the case of arbitrary camera motion, a further complication is that it is difficult to reliably track features between frames. In this section, we will describe in detail an approach to feature-based Kalman filtering for lateral camera translation that tracks edgels along each scan line and avoids nonlinear measurement equations. The restriction to lateral motion simplifies the comparison of the iconic and feature-based algorithms performed in the following section; it also has valuable practical applications in the context of manipulator-mounted cameras and in bootstrapping binocular stereo correspondence. Extensions to arbitrary motion can be based on the method presented here.

5.1. Kalman Filter Formulation for Lateral Motion

Lateral camera translation considerably simplifies the feature tracking problem, since in this case features flow along scan lines. Moreover, the position of a feature on a scan line is a linear function of the distance moved by the camera, since

$$\Delta x = -T_x \mathrm{d} \Leftrightarrow x_t = x_0 - tT_x d$$

where x_0 is the position of the feature in the first frame and d is the inverse depth of the feature. The *epipolar-plane-image* method [4] exploits these characteristics by extracting lines in "space-time" (epipolar plane) images formed by concatenating scan lines from an entire image sequence. However, sequential estimation techniques like Kalman filtering are a more practical approach to this problem because they allow images to be processing on-line by incrementally refining the depth model.

Taking x_0 and d as the state variables defining the location of the feature, instead of the 3D coordinates X and Z, keeps the entire estimation problem linear. This is advantageous because it avoids the approximations needed for error estimation with nonlinear equations. For point features, if the position of the feature in each image is given by the sequence of measurements $\tilde{\mathbf{x}} = [\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_n]^T$, knowledge of the camera position for each image allows the feature location to be determined by fitting a line to the measurement vector $\tilde{\mathbf{x}}$:

$$\tilde{\mathbf{x}} = \mathbf{H} \begin{bmatrix} x_0 \\ d \end{bmatrix} \qquad [16]$$

where $\mathbf{H}$ is a $(n + 1) \times 2$ matrix whose first column contains all 1's and whose second column is defined by the camera position for each frame, relative to the initial camera position. This fit can be computed sequentially by accumulating the terms of the normal equation solution for x_0 and d. The covariance matrix Σ of x_0 and d can be determined from the covariance matrix of the measurement vector $\tilde{\mathbf{x}}$.

The approach outlined above uses the position of the feature in the first frame x_0 as one of the two state variables. We can reformulate this in terms of the current frame by taking x_t and d to be the state variables. Assuming that the camera motion is exact and that measured feature positions have normally distributed uncertainty with variance σ_e^2, the initial state vector and covariance matrix are expressed in terms of ideal image coordinates as

$$x_1 = \tilde{x}_1$$

$$d = \frac{\tilde{x}_0 - \tilde{x}_1}{T_1}$$

$$P_0^+ = \sigma_e^2 \begin{bmatrix} 1 & -1/T_1 \\ -1/T_1 & 2/T_1^2 \end{bmatrix}$$

where T_1 is the camera translation between the first and second frame. The covariance matrix comes from applying standard linear error propagation methods to the equations for x_1 and d [19].

After initialization, if T_t is the translation between frames $t - 1$ and t, the motion equations that transform the state vector and covariance matrix to the current frame are

$$u_t^- = \begin{bmatrix} x_t^- \\ d_t^- \end{bmatrix} = \begin{bmatrix} 1 & -T_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1}^+ \\ d_{t-1}^+ \end{bmatrix} = \Phi_t u_{t-1}^+ \qquad [17]$$

$$P_t^- = \Phi_t P_{t-1}^+ \Phi_t^T \qquad [18]$$

The superscript minuses indicate that these estimates do not incorporate the measured edge position at time t. The newly measured edge position $\tilde{x}_t$ is incorporated by computing the updated covariance matrix P_t^+, a gain matrix K, and the updated parameter vector u_t^+:

$$P_t^+ = \{(P_t^-)^{-1} + S\}^{-1}$$

$$\text{where} \qquad S = \frac{1}{\sigma_e^2} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$K = \frac{1}{\sigma_t^2} P_t^+ \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$u_t^+ = u_t^- + K [\tilde{x}_t - x_t^-]$$

Since these equations are linear, we can see how uncertainty decreases as the number of measurements increases by computing the sequence of covariance matrices P_t, given only the measurement uncertainty σ_e^2 and the sequence of camera motions T_t. This is addressed in section 6.1

Note that the equations above can be generalized to arbitrary, uncertain camera motion using either the x, y, d image-based parameterization of point locations or an X, Y, Z three-dimensional parameterization. The choice of parameterization may affect the conditioning of general depth-from-motion algorithms, but we have addressed this question to date.

5.2. Feature Extraction and Matching

To implement the feature-based depth estimator, we must specify how to extract feature positions, how to estimate the noise level in these positions, and how to track features from frame to frame. For lateral motion, with image flow parallel to the scanlines, tracking edgels on each scanline is a natural implementation. Therefore, in this section we will describe how we extract edges to subpixel precision, how we estimate the variance of the edge positions, and how we track edges from frame to frame.

For one-dimensional signals, estimating the variance of edge positions has been addressed in [7]. We will review this analysis before considering the general case. In one dimension, edge extraction amounts to finding the zero crossings in the second derivative of the Gaussian-smoothed signal, which is equivalent to finding zero-crossings after convolving the image with a second derivative of Gaussian operator,

$$F(x) = \frac{d^2 G(x)}{dx^2} * I(x)$$

We assume that the image I is corrupted by white noise with variance σ_n^2. Splitting the response of the operator into that due to the signal F_s, and that due to noise, F_n, edges are marked where

$$F_s(x) + F_n(x) = 0 \qquad [19]$$

An expression for the edge variance is obtained by taking a first-order Taylor expansion of the deterministic part of the response in the vicinity of the zero crossing, then taking mean square values. Thus, if the zero crossing occurs at x_0 in the noise free signal and $x_0 + \delta x$ in the noisy signal, we have

$$F(x_0 + \delta x) \approx F_s(x_0) + F_s'(x_0)\,\delta x + F_n(x_0 + \delta x) = 0 \quad [20]$$

so that

$$\delta x = \frac{-(F_n(x_0 + \delta x) + F_s(x_0))}{F_s'(x_0)} \quad [21]$$

The presence of a zero crossing implies that $F_s(x_0) = 0$ and the assumption of zero mean noise implies that $E[F_n(x_0)] = 0$. Therefore, the variance of the edge position is

$$E[\delta x^2] = \sigma_e^2 = \frac{\sigma_n^2 E[(F_n(x_0))^2]}{(F_s'(x_0))^2} \quad [22]$$

In a discrete implementation, $E[(F_n(x_0))^2]$ is the sum of the squares of the coefficients in the convolution mask. $F_s'(x_0)$ is the slope of the zero crossing and is approximated by fitting a local curve to the filtered image. The zero crossing of this curve gives the estimate of the subpixel edge position.

For two-dimensional images, an analogous edge operator is a directional derivative filter with a derivative of Gaussian profile in one direction and a Gaussian profile in the orthogonal direction. Assuming that the operator is oriented to take the derivative in the direction of the gradient, the analysis above will give the variance of the edge position in the direction of the gradient (see [23] for an alternate approach). However, for edge tracking along scanlines, we require the variance of the edge position in the scanline direction, not the gradient direction. This is straightforward to compute for the difference of Gaussian *(DOG)* edge operator; the required variance estimate comes directly from equations (19)–(22), replacing F with the *DOG* and F' with the partial derivative $\partial/\partial x$. Details of the discrete implementation in this case are similar to those described above. Experimentally, the cameras and digitizing hardware we use provide 8-bit images with intensity variance $\sigma_n^2 \approx 4$.

It is worth emphasizing that estimating the variance of edge positions is more than a mathematical nicety; it is valuable in practice. The uncertainty in the position of an edge is affected by the contrast of the edge, the amount of noise in the image, and, in matching applications such as this one, by the edge orientiation. For example, in tracking edges under lateral motion, edges that are close to horizontal provide much less precise depth estimates than edges that are vertical. Estimating variance quantifies these differences in precision. Such quantification is important in predictive tracking, fitting surface models, and applications of depth-from-motion to constraining stereo. These remarks of course apply to image features in general, not just to edges.

Tracking features from frame to frame is very simple if either the camera motion is very small or the feature depth is already known quite accurately. In the former case, a search window is defined that limits the feature displacement to a small number of pixels from the position in the previous image. For the experiments described in section 6, tracking was implemented this way, with a window width of two pixels. Alternatively, when the depth of a feature is already known fairly accurately, the position of the feature in a new image can be predicted from equation (17) to be

$$x_t^- = x_{t-1}^+ - T_t d_{t-1}^+$$

the variance of the prediction can be determined from equation (18), and a search window can be defined as a confidence interval estimated from this variance. This allows tight search windows to be defined for existing features even when the camera motion is not small. A simplified version of this procedure is used in our implementation to ensure that candidate edge matches are consistent with the existing depth model. The predefined search window is scanned for possible matches, and these are accepted only if they lie within some distance of the predicted edge location. Additional acceptance criteria require the candidate match to have properties similar to those of the feature in the previous image; for edges, these properties are edge orientation and edge strength (gradient magnitude or zero-crossing slope). Given knowledge of the noise level in the image, this comparison function can be defined probabilistically as well, but we have not pursued this direction.

Finally, if the noise level in the image is unknown it can be estimated from the residuals of the observations after x and d have been determined. Such methods are discussed in [21] for batch-oriented techniques analogous to equation (16) and in [18] for Kalman filtering.

6 Evaluation

In this section, we compare the performance of the iconic and feature-based depth estimation algorithms in three ways. First, we perform a mathematical analysis of the reduction in depth variance as a function of time. Second, we use a sequence of images of a flat scene to determine the quantitative performance of the two approaches and to check the validity of our analysis. Third, we test our algorithms on images of realistic scenes with complicated variations in depth.

6.1. Mathematical Analysis

We wish to compare the theoretical variance of the depth estimates obtained by the iconic method of section 4 to those obtained by the feature-based method of section 5. We will also compare the accuracy of both methods to the accuracy of stereo matching with the first and last frames of the image sequence. To do this, we will derive expressions for the depth variance as a function of the number of frames processed, assuming a constant noise level in the images and constant camera motion between frames. For clarity, we will assume this motion is $T_x = 1$.

6.1.1. Iconic Approach. For the iconic method, we will ignore process noise in the system model and assume that the variance of successive flow measurements is constant. For lateral motion, the equations developed in section 2 can be simplified to show that the Kalman filter simply computes the average flow [30]. Therefore, a sequence of flow measurements $\Delta x_1, \Delta x_2, \ldots, \Delta x_t$ is equivalent to the following batch measurement equation

$$\Delta \mathbf{x} = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_t \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} d = \mathbf{H}d$$

Estimating d by averaging the flow measurements implies that

$$d = -\frac{1}{t} H^T \Delta \mathbf{x} = -\frac{1}{t} \sum_{i=1}^{t} \Delta x_i \qquad [23]$$

If the flow measurements were independent with variance $2\sigma_n^2/a$, where σ_n^2 is the noise level in the image (appendix A), the resulting variance of the disparity estimate would be

$$\frac{2\sigma_n^2}{ta} \qquad [24]$$

However, the flow measurements are not actually independent. Because noise is present in every image, flow measurements between frames $i-1$ and i will be correlated with measurements for frames i and $i+1$. Appendix A shows that a sequence of correlation-based flow measurements that track the same point in the image sequence will have the following covariance matrix:

$$P_m = \frac{\sigma_n^2}{a} \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & . & & & \\ & & & . & & \\ & & & & . & \\ & & & & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}$$

where σ_n^2 is the level of noise in the image and a reflects the local slope of the intensity surface. With this covariance matrix, averaging the flow measurements actually yields the following variance for the estimated flow:

$$\sigma_f^2(t) = \frac{1}{t^2} H^T P_m H = \frac{2\sigma_n^2}{t^2 a} \qquad [25]$$

This is interesting and rather surprising. Comparing equations (24) and (25), the correlation structure that exists in the measurements means that the algorithm converges faster than we first expected.

With correlated measurements, averaging the flow measurements in fact is a suboptimal estimator for d. The optimal estimator is obtained by substituting the expressions for H and P_m into equation (12) and (13). This estimator does not give equal weight to all flow measurements; instead, measurements near the center of the sequence receive more weight than those near the end. The variance of the depth estimate is

$$\sigma_{f'}^2(t) = \frac{12\sigma_n^2}{t(t+1)(t+2)a}$$

The optimal convergence is cubic, whereas the convergence of the averaging method we implemented is quadratic. Developing an incremental version of the optimal estimator requires extending our Kalman filter formulation to model the correlated nature of the measurements. This extension is currently being investigated.

6.1.2. Feature-based approach. For the feature-based approach, the desired variance estimates come from

computing the sequence of covariance matrices P_t, as mentioned at the end of section 5.1. A closed form expression for this matrix is easier to obtain from the batch method suggested by equation (16) than from the Kalman filter formulation and yields an equivalent result. Taking the constant camera translation to be $T_x = 1$ for simplicity, equation (16) expands to

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x}_0 \\ \tilde{x}_1 \\ \cdot \\ \cdot \\ \cdot \\ \tilde{x}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & -t \end{bmatrix} \begin{bmatrix} x_0 \\ d \end{bmatrix} = H\mathbf{u} \qquad [26]$$

Recall that $\tilde{x}_i$ are the edge positions in each frame, x_0 is the best fit edge position in the first frame, and d is the best fit displacement or flow between frames. Since we assume that the measured edge positions $\tilde{x}_i$ are independent with equal variance σ_e^2, we find that

$$P_F = \begin{bmatrix} \sigma_x^2 & \sigma_{xd} \\ \sigma_{xd} & \sigma_d^2 \end{bmatrix} = \sigma_e^2 \begin{bmatrix} \sum_{i=0}^{t} 1 & -\sum_{i=0}^{t} i \\ -\sum_{i=0}^{t} i & \sum_{i=0}^{t} i^2 \end{bmatrix}^{-1} \qquad [27]$$

The summations can be expressed in closed form, leading to the conclusion that

$$\sigma_F^2(t) = \frac{12\sigma_e^2}{t(t+1)(t+2)} \qquad [28]$$

The variance of the displacement or flow estimate d thus decreases as the cube of the number of images. This expression is identical in structure to the optimal estimate for the iconic approach, the only difference being the replacement of the variable of the SSD minimum by the variance of the edge position. Thus, if our estimators incorporate appropriate models of measurement noise, the iconic and feature-based methods theoretically achieve the same rate of convergence. This is surprising, given that the basic Kalman filter for the iconic method maintains only one state parameter *(d)* for each pixel, whereas the feature-based method maintains two per feature (x_0 and *d)*. We suspect that an incremental version of the optimal iconic estimator will require the same amount of state as the feature-based method.

6.1.3. Comparison with Stereo. To compare these methods to stereo matching on the first and last frames of the image sequence, we must scale the stereo disparity and its uncertainty to be commensurate with the flow between frames. This implies dividing the stereo disparity by t and the uncertainty by t^2. For the iconic method, we assume that the uncertainty in a stereo measurement will be the same as that for an individual flow measurement. Thus, the scaled uncertainty is

$$\sigma_{IS}^2(t) = \frac{2\sigma_n^2}{t^2 a}$$

This is the same as is achieved with our incremental algorithm which processes all of the intermediate frames. Therefore, processing the intermediate frames (while ignoring the temporal correlation of the measurements) may improve the reliability of the matching but in this case it does not improve precision.

For the feature-based approach, the uncertainty in stereo disparity is twice the uncertainty σ_e^2 in the feature position; the scaled uncertainty is therefore

$$\sigma_{FS}^2(t) = \frac{2\sigma_e^2}{t^2}$$

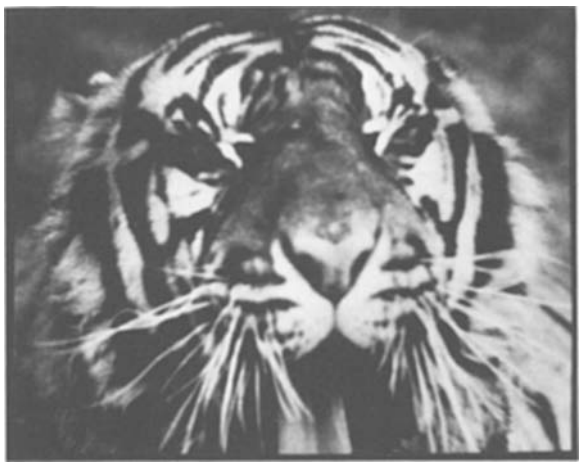
In this case using the intermediate frames helps, since

$$\frac{\sigma_F(t)}{\sigma_{FS}(t)} = \frac{1}{O(\sqrt{t})}$$

Thus, extracting depth from a small-motion image sequence has several advantages over stereo matching between the first and last frames. The ease of matching is increased, reducing the number of correspondence errors. Occlusion is less of a problem, since it can be predicted from early measurements. Finally, better accuracy is available by using the feature-based method or the optimal version of the iconic method.

6.2. Quantitative Experiments: Flat Images

The goals of our quantitative evaluation were to examine the actual convergence rates of the depth estimators, to assess the validity of the noise models, and to compare the performance of the iconic and feature-based algorithms. To obtain ground truth depth data, we used the facilities of the Calibrated Imaging Laboratory at CMU to digitize a sequence of images of a flat-mounted poster. We used a Sony XC-37 CCD camera with a 16-mm lens, which gave a field of view of 36 degrees. The poster was set about 20 inches (51 cm) from the camera. The camera motion between frames was 0.04



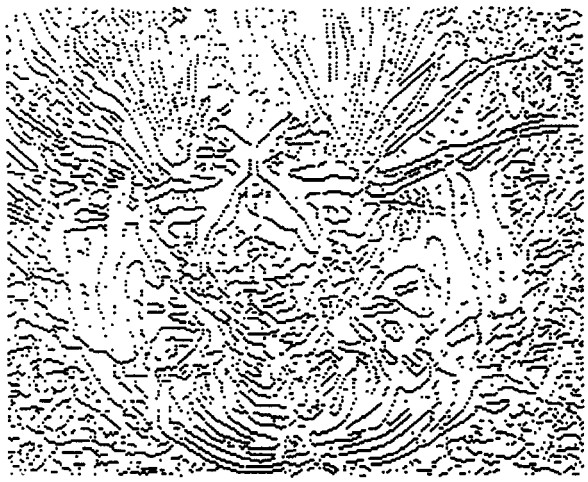


Fig. 8. Tiger image and edges.

inches (1 mm), which gave an actual flow of approximately two pixels per frame in 480×512 images. For convenience, our experiments were run on images reduced to 240×256 by Gaussian convolution and subsampling. The image sequence we will discuss here was taken with vertical camera motion. This proved to give somewhat better results than horizontal motion; we attribute this to jitter in the scanline clock, which induces more noise in horizontal flow than in vertical flow.

Figure 8 shows the poster and the edges extracted from it. For both the iconic and the feature-based algorithms, a ground truth value for the depth was determined by fitting a plane to the measured values. The level of measurement noise was then estimated by computing the RMS deviation of the measurements from the plane fit. Optical aberrations made the flow measurements consistently smaller near the periphery of the image than the center, so the RMS calculation was performed over only the center quarter of the image. Note that all experiments described in this section did *not* use regularization to smooth the depth estimates, so the results show only the effect of the Kalman filtering algorithm.

To determine the reliability of the flow variance estimates, we grouped flow measurements produced by the SSD algorithm according to their estimated variances, took sample variances over each group, and plotted the SSD variance estimates against the sample variances (figure 9). The strong linear relationship indicates fairly reliable variance estimates. The deviation of the slope of the line from the ideal value of one is due to an inaccurate estimate of the image noise (σ_n^2).

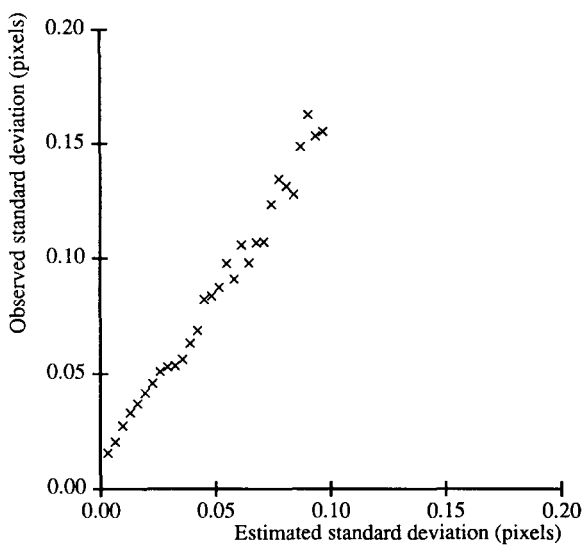


Fig. 9 Scatter plot.

To examine the convergence of the Kalman filter, the RMS depth error was computed for the iconic and the feature-based algorithms after processing each image in the sequence. We computed two sets of statistics, one for "sparse" depth and one for "dense" depth. The sparse statistic computes the RMS error for only those pixels where both algorithms gave depth estimates (that is, where edges were found), whereas the dense statistic computes the RMS error of the iconic algorithm over the full image. Figure 10 plots the relative RMS errors as a function of the number of images processed. Comparing the sparse error curves, the convergence rate of

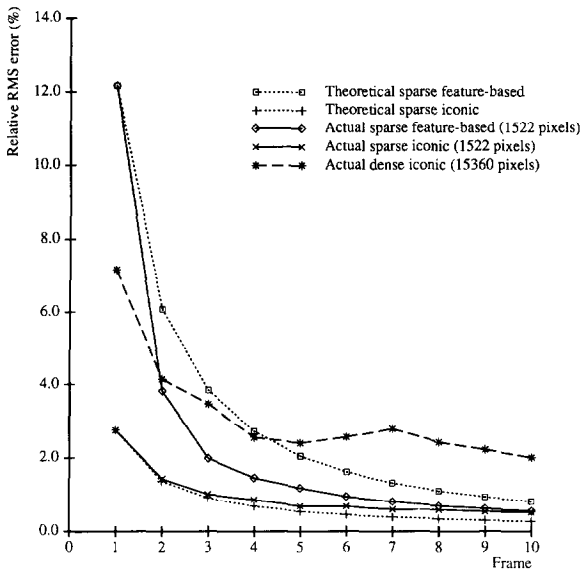


Fig. 10. RMS error in depth estimate.

the iconic algorithm is slower than the feature-based algorithm, as expected. In this particular experiment, both methods converged to an error level of approximately 0.5% after processing eleven images. Since the poster was 20 inches from the camera, this equates to a depth error of 0.1 inches. Note that the overall baseline between the first and the eleventh image was only 0.44 inches.

To compare the theoretical convergence rates derived earlier to the experimental rates, the theoretical curves were scaled to coincide with the experimental error after processing the first two frames. These scaled curves are also shown in figure 10. For the iconic method, the theoretical rate plotted is the quadratic convergence predicted by the correlated flow measurement model. The agreement between theory and practice is quite good for the first three frames. Thereafter, the experimental RMS error decreases more slowly; this is probably due to the effects of unmodeled sources of noise. For the feature-based method, the experimental error initially decreases faster than predicted because the implementation required new edge matches to be consistent with the prior depth estimate. When this requirement was dropped, the results agreed very closely with the expected convergence rate.

Note that the comparison between theoretical and experimental results also allows us to estimate the precision of the subpixel edge extractor. The variance of a disparity estimate is twice the variance of the edge positions. Since the frame-to-frame displacement in this image sequence was one pixel and the relative RMS error was 12% for the first disparity estimate, the RMS error in edge localization was $0.12/\sqrt{2} \approx 0.09$ pixels.

Finally, Figure 10 also compares the RMS error for the sparse and dense depth estimates from the iconic method. The dense flow field is considerably noisier than the flow estimates that coincide with edges, though still just over two percent error by the end of eleven frames. Some of this error is due to a systematic bias produced by the SSD flow estimator in the vicinity of ramp edges. This is illustrated in figure 11. Figure 11a shows a test image of horizontal bars and figure 11b shows an intensity profile of a vertical slice taken through one of the light-to-dark transitions. Disparity and variance estimates computed along this profile are shown in figures 11c and 11d, respectively. As can be seen, the disparity estimate is biased low (away from the "true" value in the central flat part) on one side of the discontinuity, and biased "high" on the other. This bias can also be confirmed by using an analytic model of a ramp edge. Fortunately, the variance estimates reflect this large error, so regularization-based smoothing can compensate for this systematic error. We conclude that the dense depth estimates do provide fairly good depth information.

6.3. Qualitative Experiments: Real Scenes

We have also tested the iconic and edge-based algorithms on complicated, realistic scenes obtained from the Calibrated Imaging Laboratory. Two sequences of ten images were taken with camera motion of 0.05 inches (1.27 mm) between frames; one sequence moved the camera vertically, the other horizontally. The overall range of motion was therefore 0.5 inches (1.27 cm); this compares with distances to objects in the scene of 20 to 40 inches (51 to 102 cm).

Figure 12 shows one of the images. Figures 13a–d show a reduced version of the image, the edges extracted from it with an oriented Canny operator [7], and depth maps produced by applying the iconic algorithm to the horizontal and vertical image sequences, respectively. Lighter areas in the depth maps are nearer. The main structure of the scene is recovered quite well in both cases, though the results with the horizontal sequence are considerably more noisy. This is most likely due to scanline jitter, as mentioned earlier. Edges oriented parallel to the direction of flow cause some scene structure to be observable in one sequence but not the other. This is most noticeable near the center of the scene, where a thin vertical object appears in

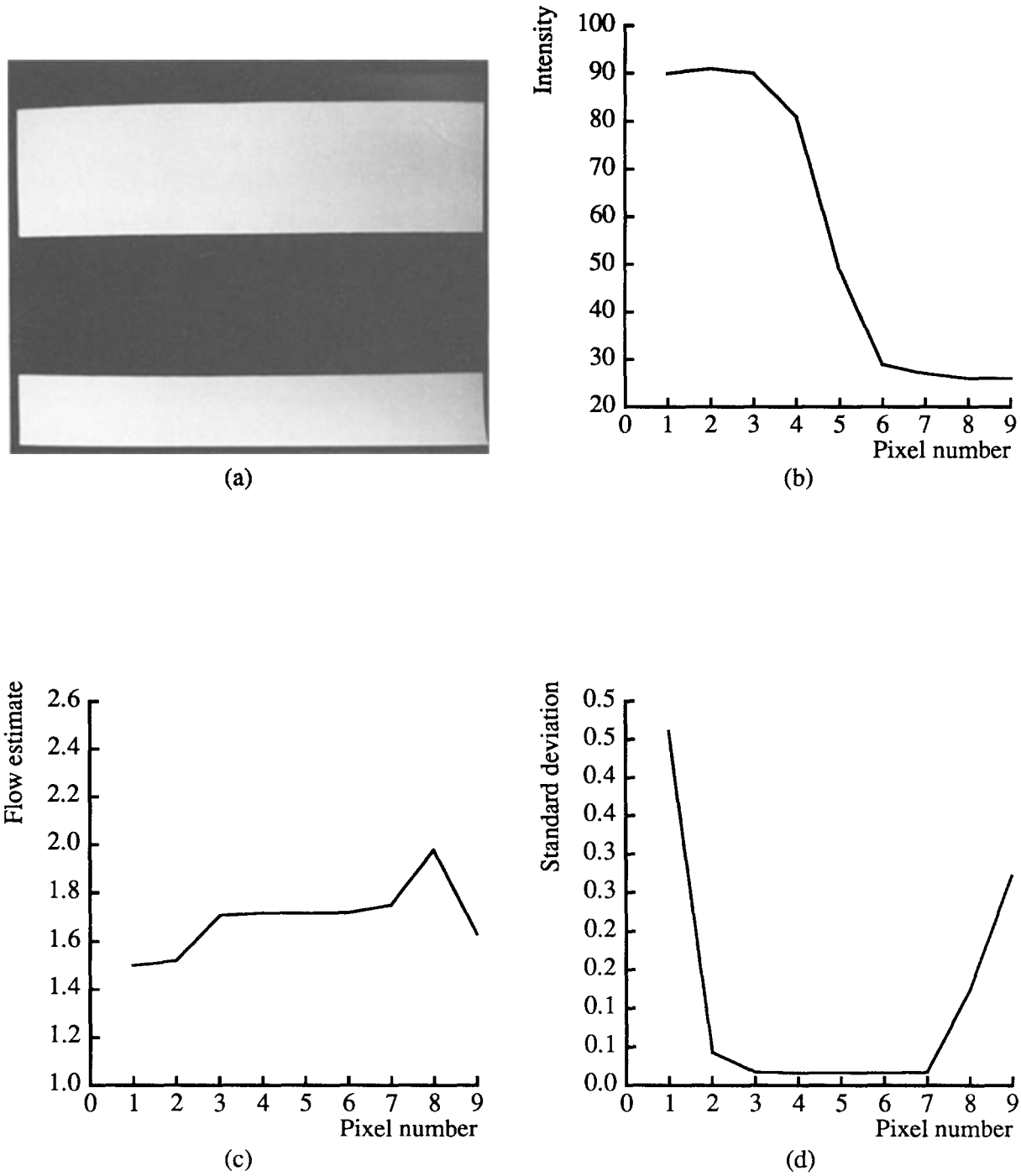


Fig 11. Bias of subpixel correlations: (a) test image, (b) intensity profile acorss a light-to-dark transition, (c) estimated flow along the intensity profile, (d) estimated variance along the profile.

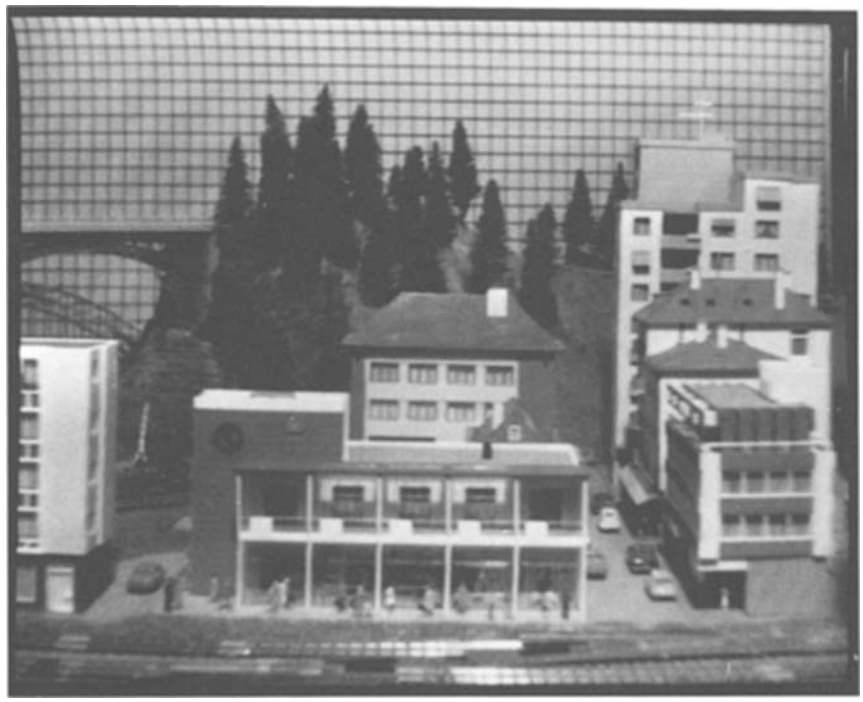


Fig. 12. CIL image.

figure 13c but is not visible in figure 13d. This object corresponds to an antenna on the top of a foreground building (figure 13a). In general, motion in orthogonal directions will yield more information than motion in any single direction.

Figure 14 shows intensity-coded depth maps and 3D perspective reconstructions obtained with both the iconic and feature-based methods. These results were produced by combining disparity estimates from both horizontal and vertical camera motion. The depth map for the feature-based approach was produced from the sparse depth estimates by regularization. It is difficult to make quantitative statements about the performance of either method from this data, but qualitatively it is clear that both recover the structure of the scene quite well.

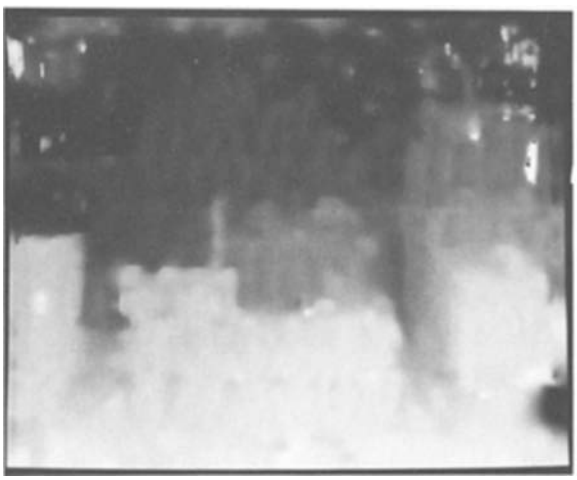
The iconic algorithm was also used to extract occluding boundaries from the depth map of figure 13c (iconic method with vertical camera motion). We first computed an intrinsic "grazing angle" image giving the angle between the view vector through each pixel and the normal vector of the local 3D surface. Edge detection and thresholding were applied to this image to find pixels where the view vector and the surface normal were nearly perpendicular. The resulting boundaries are shown along with the depth map in figure 15. The method found most of the prominent building outlines and the outline of the bridge in the upper left.

Figures 16 and 17 show the results of our algorithms on a different model set up in the Calibrated Imaging Laboratory. The same camera and camera motion were used as before. Figure 16 shows the first frame, the extracted edges, and the depth maps obtained from horizontal and vertical motion. Figure 17 shows the depth maps and the perspective reconstructions obtained with the iconic and feature-based methods. Again, the algorithms recovered the structure of the scene quite well.

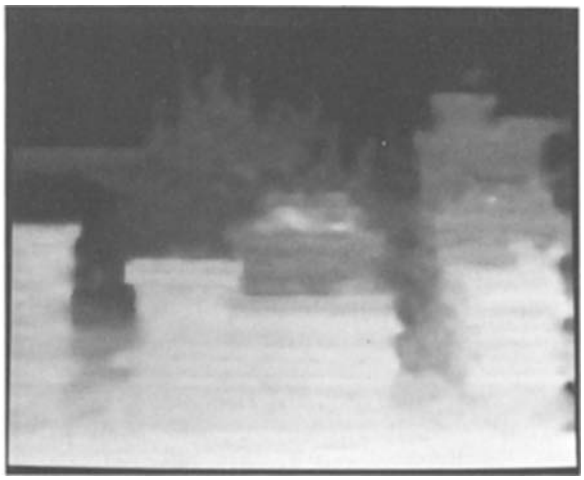
Finally, we present the results of using the first 10 frames of the image sequence used in [4]. Figure 18 shows the first frame from the sequence, the extracted edges, and the depth maps obtained from running the iconic and feature-based algorithms. As expected, the results from using the feature-based method are similar to those obtained with the epipolar-plane image technique [4]. The iconic algorithm produces a denser estimate of depth than is available from either edge-based technique. These results show that the sparse (edge-based) batch-processing algorithm for small motion sequences introduced in [4] can be extended to use dense depth maps and incremental processing.

(a)

(b)



(c)



(d)

Fig 13. CIL depth maps: (a) first frame, (b) edges, (c) horizontal-motion depth map, (d) vertical-motion depth map.

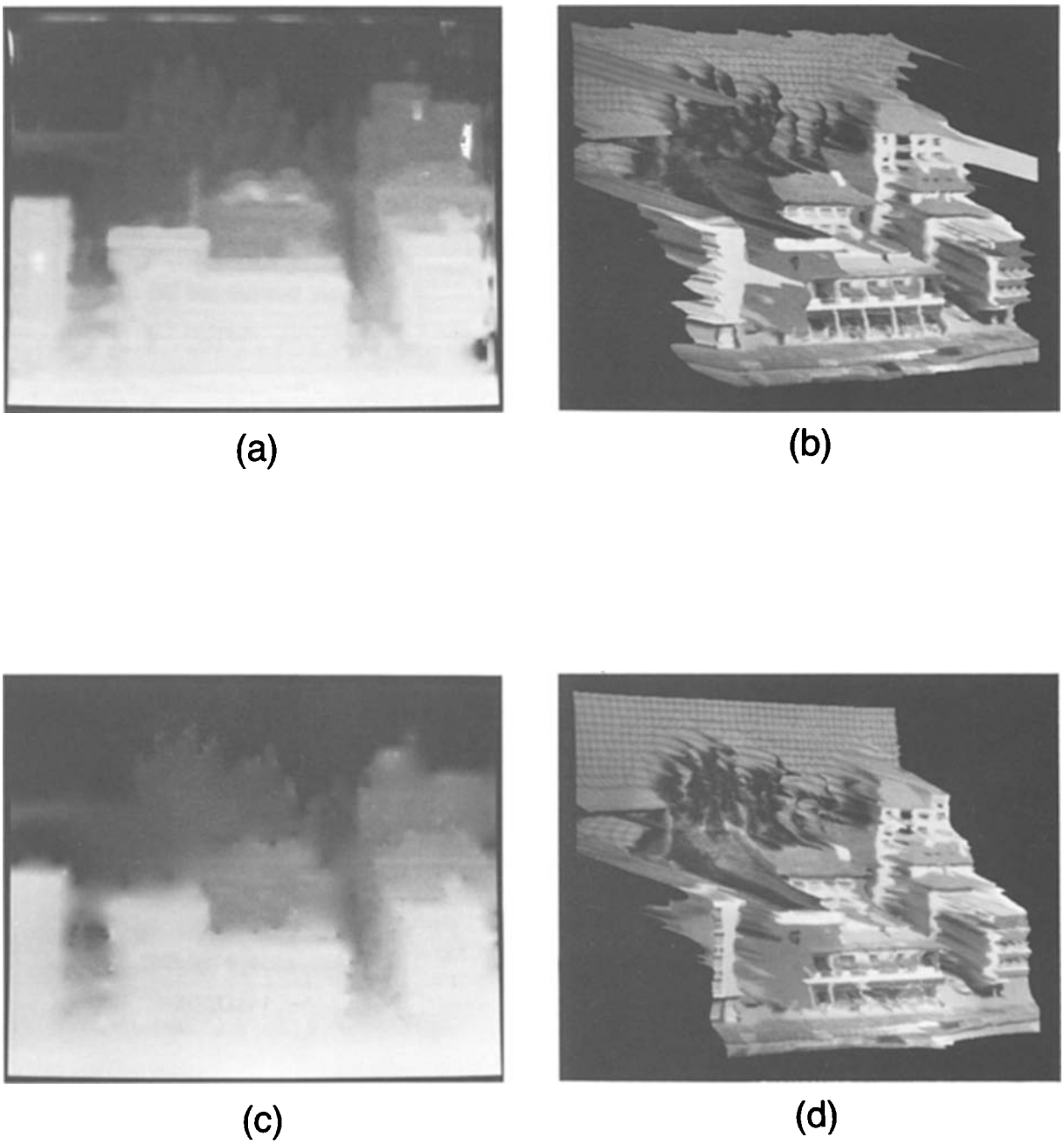


Fig. 14. CIL orthogonal motion results: (a) iconic method depth map, (b) perspective view, (c) feature-based method depth map, (d) perspective view.

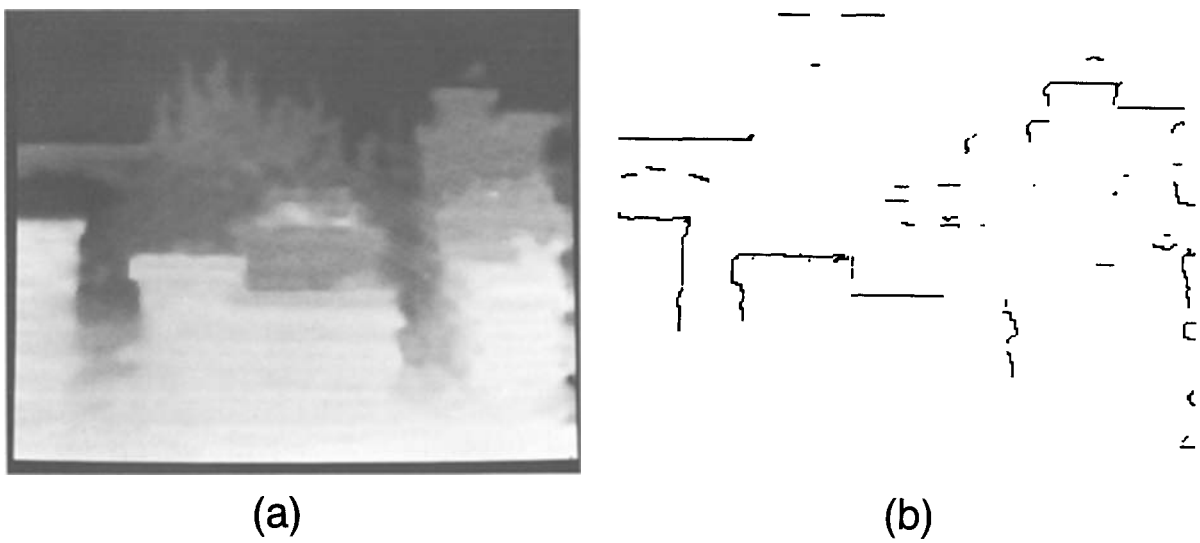
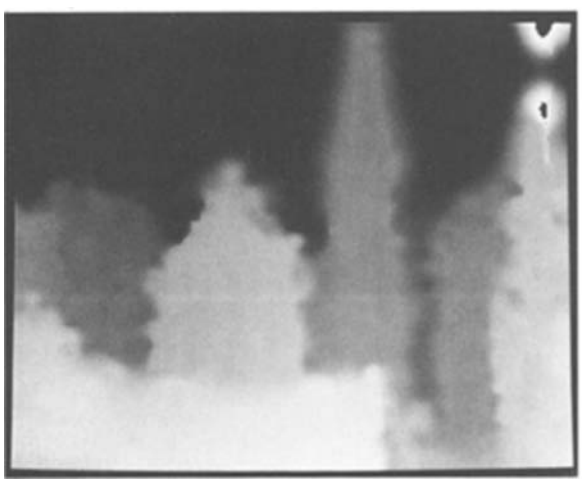


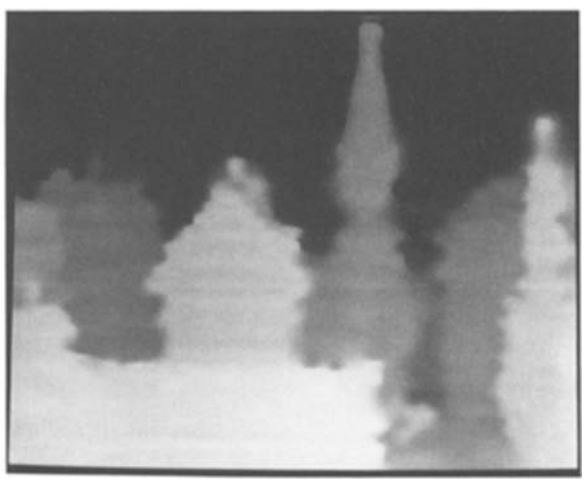
Fig. 15 Occluding boundaries. (a) vertical-motion depth map, (b) occluding boundaries

(a)

(b)



(c)



(d)

Fig. 16. CIL-2depth maps: (a) first frame, (b) edges, (c) horizontal-motion depth map, (d) vertical-motion depth map.

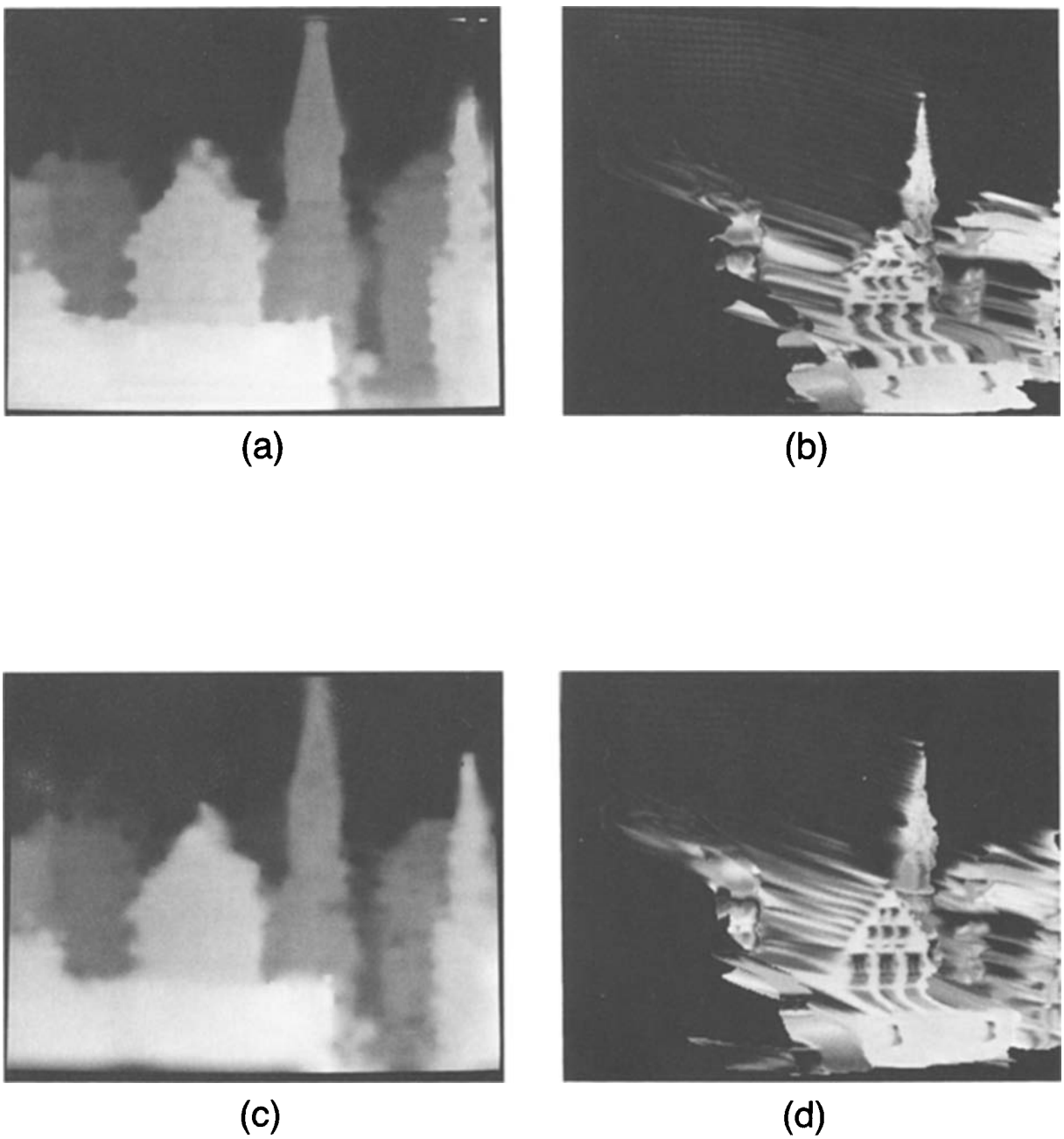
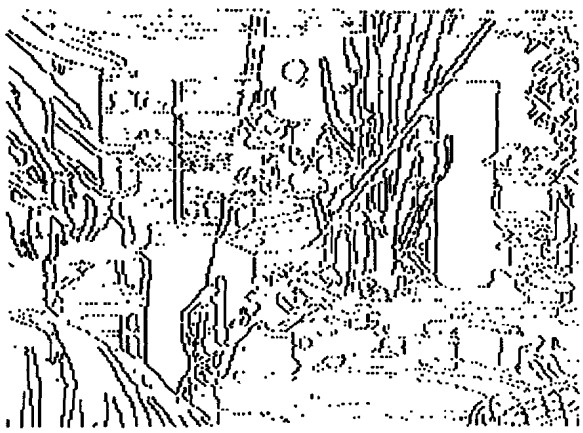
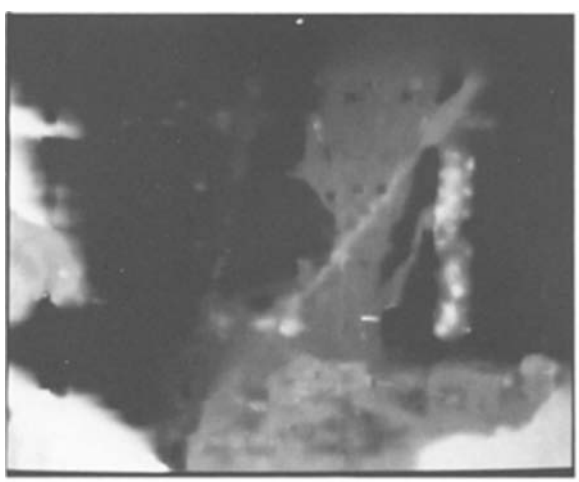


Fig 17. CIL-2 orthogonal motion results: (a) iconic method depth map, (b) perspective view, (c) feature-based method depth map, (d) perspective view

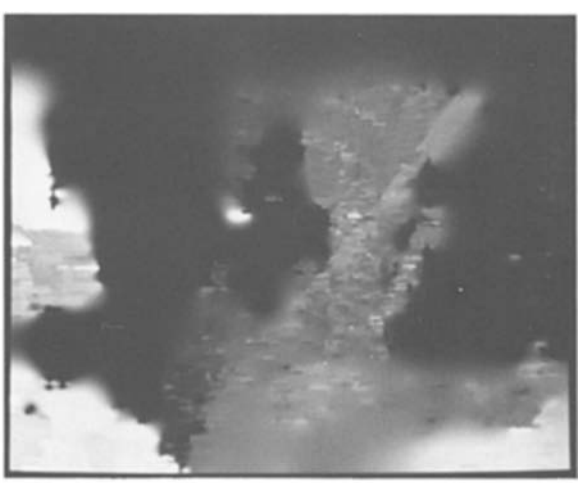
(a)



(b)



(c)



(d)

Fig. 18. SRI EPI sequence results: (a) first frame, (b) edges, (c) iconic method depth map, (d) feature-based method depth map.

7 Conclusions

This paper has presented a new algorithm for extracting depth from known motion. The algorithm processes an image sequence taken with small interframe displacements and produces an on-line estimate of depth that is refined over time. The algorithm produces a dense, iconic depth map and is suitable for implementation on parallel architectures.

The on-line depth estimator is based on Kalman filtering. A correlation-based flow algorithm measures both the local displacement at each pixel and the confidence (or variance) of the displacement. These two "measurement images" are integrated with predicted depth and variance maps using a weighted least-squares technique derived from the Kalman filter. Regularization-based smoothing is used to fill in areas of unknown disparity and to reduce the noise in the flow estimates. The current maps are extrapolated to the next frame by image warping, using knowledge of the camera motion, and are resampled to keep the maps iconic.

The algorithm has been implemented for lateral camera translation, evaluated mathematically and experimentally, and compared with a feature-based algorithm that uses Kalman filtering to estimate the depth of edges. The mathematical analysis shows that the iconic approach will have a slower convergence rate because it only keeps one element of state per pixel (the disparity), while the feature-based approach keeps both the disparity and the subpixel position of the feature. However, an optimal implementation of the iconic method (which takes into account temporal correlations in the measurements) has the potential to equal the convergence rate and accuracy of the symbolic method. Experiments with images of a flat poster have confirmed this analysis and given quantitative measures of the performance of both algorithms. Finally, experiments with images of a realistic outdoor-scene model have shown that the new algorithm performs well on images with large variations in depth and that occluding boundaries can be extracted from the resulting depth maps.

The lateral motion implementations developed in this paper have several potential robotics applications. For example, manipulators with cameras mounted near the end-effector can use small translations to acquire shape information about a workpiece. In addition, binocular stereo systems can take great advantage of such motions by using depth from motion to constrain binocular correspondence. The degrees of freedom necessary for this are already available in manipulator-mounted systems and can be designed into vehicle-mounted stereo systems.

Extensions

The algorithms described in this paper can be extended in several ways. The most straightforward extension is to the case of nonlateral motion. As sketched in section 4, this can be accomplished by designing a correlation-based flow estimator that produces two-dimensional flow vectors and an associated covariance matrix estimate [1]. This approach can also be used when the camera motion is uncertain, or when the camera motion is variable (e.g., for widening baseline stereo [34]). The alternative of searching only along epipolar lines during the correlation phase may be easier to implement, but is less general.

More research is required into the behavior of the correlation based flow and confidence estimator. In particular, we have observed that our current estimator produces biased estimates in the vicinity of intensity step edges. The correlation between spatially adjacent flow estimates, which is currently ignored, should be integrated into the Kalman filter framework. More sophisticated representations for the intensity and depth fields are also being investigated [28].

Finally, as noted above, the incremental depth from motion algorithms can be used to initiate stereo fusion. Work is currently in progress investigating the integration of depth-from-motion and stereo [15]. We believe that the framework presented in this paper will prove to be useful for integrating information from multiple visual sources and for tracking such information in a dynamic environment.

Acknowledgement

This research was sponsored in part by DARPA, monitored by the Air Force Avionics Lab under contract F33615-87-C-1499 and in part by a postgraduate fellowship from the FMC Corporation. Data for this research was partially provided by the Calibrated Imaging Laboratory at CMU. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

Appendix A: Optic Flow Computation

In this appendix, we will analyze the performance of a simple correlation-based flow estimator, the sum-of-

squared-differences (SSD) estimator [1]. This estimator selects at each pixel the disparity that minimizes the SSD measure

$$e(\tilde{d};x) = \int w(\lambda)[f_1(x + \tilde{d} + \lambda) - f_0(x + \lambda)]^2 \, d\lambda$$

where $f_0(x)$ and $f_1(x)$ are the two successive image frames, and $w(\lambda)$ is a symmetric, non-negative weighting function. To analyze its performance, we will assume that the two image frames are generated from an underlying true intensity image, $f(x)$, to which uncorrelated (white) Gaussian noise with variance σ_n^2 has been added:

$$f_0(x) = f(x) + n_0(x),$$
$$f_1(x) = f(x - d) + n_1(x)$$

Using this model, we can rewrite the error measure as*

$$e(\tilde{d};x) = \int w(\lambda)[f(x + \tilde{d} - d + \lambda) - f(x + \lambda) + n_1(x + \lambda) - n_0(x + \lambda)]^2 \, d\lambda$$

If $\tilde{d} \simeq d$, we can use a Taylor series expansion to obtain

$$\begin{aligned} e(\tilde{d};x) &= \int w(\lambda)[f'(x + \lambda)](\tilde{d} - d)^2 \\ &\quad + 2w(\lambda)f'(x + \lambda) \\ &\quad \times [n_1(x + \lambda) - n_0(x + \lambda)] \\ &\quad \times (\tilde{d} - d) + w(\lambda) \\ &\quad \times [n_1(x + \lambda) - n_0(x + \lambda)]^2 \, d\lambda \\ &= a(x)(\tilde{d} - d)^2 + 2[b_1(x) - b_0(x)] \\ &\quad \times (\tilde{d} - d) + c(x) \end{aligned}$$

where

$$a(x) = \int w(\lambda)[f'(x + \lambda)]^2 \, d\lambda$$

$$b_i(x) = \int w(\lambda)f'(x + \lambda)n_i(x + \lambda) \, d\lambda$$

$$c(x) = \int w(\lambda)[n_1(x + \lambda) - n_0(x + \lambda)]^2 \, d\lambda$$

*This equation is actually incorrect, since it should contain $n_1(x + \tilde{d} - d + \lambda)$ instead of $n_1(x + \lambda)$. The effect of including the correct term is to add small random terms involving integrals of $w(\lambda)$, $w'(\lambda)$, $f'(x + \lambda)$, $f''(x + \lambda)$, and $n_1(x)$ to the quadratic coefficients $a(x)$, $b_1(x)$, and $c(x)$ that are derived below. This intentional omission has been made to simplify the presentation.

The four coefficients $a(x)$, $b_0(x)$, $b_1(x)$, and $c(x)$ define the shape of the error surface $e(\tilde{d};x)$. The first coefficient, $a(x)$, is related to the average "roughness" or "slope" of the intensity surface, and determines the confidence given to the disparity estimate (see below). The second and third coefficients, $b_0(x)$ and $b_1(x)$, are independent, zero-mean Gaussian random variables that determine the difference between $\hat{d}$ and d, i.e., the error in flow estimator. The fourth coefficient, $c(x)$, is a chi-squared-distributed random variable with mean $(2\sigma_n^2 \int w(\lambda) \, d\lambda)$, and defines the computed error at $\tilde{d} = d$.

To estimate the disparity at point x given the error surface $e(\tilde{d};x)$, we find the $\hat{d}$ such that

$$e(\hat{d};x) = \min_{\tilde{d}}(\tilde{d};x)$$

From the above quadratic equation,‡ we can compute $\hat{d}(x)$ as

$$\hat{d}(x) = d + \frac{b_0(x) - b_1(x)}{a(x)}$$

To calculate the variance in this estimate, we must first calculate the variance in $b_i(x)$,

$$\text{var}\,(b_i(x)) = \sigma_n^2 \int w^2(\lambda)[f'(x + \lambda)]^2 \, d\lambda$$

If we set $w(x) = 1$ on some finite interval, and zero elsewhere, this variance reduces to $\sigma_n^2 a(x)$, and we obtain

$$\text{var}\,(\hat{d}) = \frac{2\sigma_n^2}{a(x)}$$

In addition to calculating the disparity-estimate variance, we can compute its covariance with other estimates either in the same frame or in a subsequent frame. As described in section 6.1, knowing the correlation between adjacent or successive measurements is important in obtaining good overall uncertainty estimates.

To determine the correlation between two adjacent disparity estimates, $\hat{d}(x)$ and $\hat{d}(x + \Delta x)$, we must first determine the correlation between $b_i(x)$ and $b_i(x + \Delta x)$,

$$\begin{aligned} &\langle b_i(x)b_i(x + \Delta x)\rangle \\ &\quad = \int\int w(\lambda)w(\eta)f'(x + \lambda)f'(x + \Delta x + \eta) \\ &\qquad \times \langle n_i(x + \lambda)n_i(x + \Delta x + \eta)\rangle \, d\lambda \, d\eta \end{aligned}$$

‡The true equation (when higher order Taylor series terms are included) is a polynomial series in $(\tilde{d}-d)$ with random coefficients of decreasing variance. This explains the "rough" nature of the $e(\tilde{d};x)$ observed in practice.

$$= \int\int w(\lambda)w(\eta)f'(x + \lambda)f'(x + \Delta x + \eta) \times \delta(\lambda - \Delta x - \eta)\sigma_n^2 \, d\lambda \, d\eta$$

$$= \sigma_n^2 \int w(\lambda)w(\lambda - \Delta x)[f'(x + \lambda)]^2 \, d\lambda$$

For a slowly varying gradient $f'(x)$, this correlation is proportional to the autocorrelation of the weighting function,

$$R_w(\Delta x) = \int w(\lambda)w(\lambda + \Delta x) \, d\lambda$$

For the simple case of $w(\lambda) = 1$ on $[-s, s]$, we obtain

$$R_{\hat{d}}(x, x + \Delta x) = \frac{2\sigma_n^2}{a(x)}\left(1 - \frac{|x|}{2s}\right) \quad \text{for } |x| \leq 2s$$

The correlation between two successive measurements in time is easier to compute. Since

$$f_2(x + 2d) = f(x) + n_2(x)$$

we can show that the flow estimate obtained from the second pair of frames is

$$\hat{d}_2(x) = d + \frac{b_2(x) - b_1(x)}{a(x)}$$

The covariance between $\hat{d}_1(x)$ and $\hat{d}_2(x)$ is

$$\text{cov}(\hat{d}_1(x), \hat{d}_2(x)) = \langle(\hat{d}_1(x) - d)(\hat{d}_2(x) - d)\rangle = -\frac{\sigma_n^2}{a(x)}$$

and the covariance matrix of the sequence of measurements $\hat{d}_t$ is

$$P_m = \frac{\sigma^2}{a}\begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & \cdot & & & \\ & & & \cdot & & \\ & & & & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}$$

This structure is used in section 6.1 to estimate the theoretical accuracy and convergence rate of the iconic depth from motion algorithm.

Appendix B: Three-Dimensional Discontinuity Detection

To calculate a discontinuity in the depth map, we compute the angle between the local normal **N** and the view vector **V**. The surface normal at pixel value (r, c) is computed by using the 3D locations of the three points:

$$\mathbf{P}_0 = (X_0, Y_0, Z_0) = (x_0, y_0, 1)\frac{T_x}{d_0}$$

where

$$x_0 = \frac{c - c_x}{s_x}, y_0 = -\frac{r - c_y}{s_y}$$

$$\mathbf{P}_1 = (X_1, Y_1, Z_1) = (x_1, y_1, 1)\frac{T_x}{d_1}$$

where

$$x_1 = \frac{c + 1 - c_x}{s_x} = x_0 + \frac{1}{s_x},$$

$$y_1 = -\frac{r - c_y}{s_y} = y_0$$

$$\mathbf{P}_2 = (X_2, Y_2, Z_2) = (x_2, y_2, 1)\frac{T_x}{d_2}$$

where

$$x_2 = \frac{c - c_x}{s_x} = x_0,$$

$$y_2 = -\frac{r + 1 - c_y}{s_y} = y_0 - \frac{1}{s_y}$$

We can obtain the normal from the cross product of the two vectors

$$\mathbf{Q}_1 = \mathbf{P}_1 - \mathbf{P}_0 = T_x \left[\frac{1}{s_x d_1} + x_0 \left(\frac{1}{d_1} - \frac{1}{d_0} \right), y_0 \left(\frac{1}{d_1} - \frac{1}{d_0} \right), \left(\frac{1}{d_1} - \frac{1}{d_0} \right) \right]$$

$$= \frac{T_x}{d_0 d_1} \left[\frac{d_0}{s_x} - x_0(d_1 - d_0), -y_0(d_1 - d_0), -(d_1 - d_0) \right]$$

$$\mathbf{Q}_2 = \mathbf{P}_2 - \mathbf{P}_0 = T_x \left[x_0 \left(\frac{1}{d_2} - \frac{1}{d_0} \right), -\frac{1}{s_y d_2} + y_0 \left(\frac{1}{d_2} - \frac{1}{d_0} \right), \left(\frac{1}{d_2} - \frac{1}{d_0} \right) \right]$$

$$= \frac{T_x}{d_0 d_2} \left[-x_0(d_2 - d_0), -\frac{d_0}{s_y} - y_0(d_2 - d_0), -(d_2 - d_0) \right]$$

$$\mathbf{Q}_1 \times \mathbf{Q}_2 \propto \left[-\frac{d_0(d_1 - d_0)}{s_y}, \frac{d_0(d_2 - d_0)}{s_x}, -\frac{d_0^2}{s_x s_y} + \frac{x_0 d_0(d_1 - d_0)}{s_y} - \frac{y_0 d_0(d_2 - d_0)}{s_x} \right]$$

Simplifying we obtain

$$\mathbf{N} = (-s_x \Delta_1, s_y \Delta_2, -d_0 + x_0 s_x \Delta_1 - y_0 s_y \Delta_2)$$

$$\mathbf{V} = (x_0, y_0, 1)$$

$$\mathbf{N} \cdot \mathbf{V} = -d_0$$

$$\cos\theta = \frac{\mathbf{N} \cdot \mathbf{V}}{|\mathbf{N}||\mathbf{V}|}$$

where $\Delta_1 = d_0(d_1 - d_0)$ and $\Delta_2 = d_0(d_2 - d_0)$. To implement the edge detector, we require that

$$\cos\theta < \cos\theta_t$$

or

$$[s_x^2 \Delta_1^2 + s_y^2 \Delta_2^2 + (-d_0 + x_0 s_x \Delta_1 - y_0 s_y \Delta_2)^2] \times (x_0^2 + y_0^2 + 1) > d_0^2 \sec^2\theta_t$$

If the field of view of the camera is small, we have near orthographic projection, and the above equations simplify to

$$\mathbf{N} = \left[-\frac{s_x \Delta_1}{d_0}, \frac{s_y \Delta_2}{d_0}, -1 \right] = (p, q, -1)$$

$$\mathbf{V} = (x_0, y_0, 1)$$

and this reduces to the familiar gradient-based threshold

$$p^2 + q^2 > \tan^2\theta_t$$

Appendix C: Prediction Equations

To predict the new disparity map and variance map from the current maps, we will first map each pixel to its new location and value, and then use interpolation to resample the map. For simplicity, the development given here shows only the one-dimensional case, i.e., disparity d as a function of x. The extension to two dimensions is straightforward.

The motion equations for a point in the pixel map (x, d) are

$$x' = x + t_x d + r_x$$

$$d' = d + t_z$$

We will assume that the points which define the patch under consideration have the same t_x, r_x, and t_z values. These three parameters are actually stochastic variables, due to the uncertainty in camera motion. For the lateral-motion case, we assume that the mean of t_x is known and nonzero, while the means of r_x and t_z are zero.

We can write the vector equations for the motion of the points in a patch as

$$\mathbf{x}' = \mathbf{x} + t_x \mathbf{d} + r_x \mathbf{e}$$

$$\mathbf{d}' = \mathbf{d} + t_z \mathbf{e}$$

where

$$\mathbf{x} \sim N(\hat{\mathbf{x}}, \Sigma_x), \quad t_x \sim N(\hat{t_x}, \sigma_{t_x}^2), \quad r_x \sim N(0, \sigma_{r_x}^2)$$

$$\mathbf{d} \sim N(\hat{\mathbf{d}}, \Sigma_d), \quad r_z \sim N(0, \sigma_{t_z}^2),$$

$$\mathbf{e} = [1 \ldots 1]^T$$

The Jacobian of this vector equation is

$$\frac{\partial(\mathbf{x}', \mathbf{d}')}{\partial(\mathbf{x}, \mathbf{d}, t_x, r_x, t_z)} = \begin{bmatrix} \mathbf{I} & t_x \mathbf{I} & \mathbf{d} & \mathbf{e} & 0 \\ 0 & \mathbf{I} & 0 & 0 & \mathbf{e} \end{bmatrix}^T$$

and the variance of the predicted points is

$$\text{var}(\mathbf{x}', \mathbf{d}') = \begin{bmatrix} \Sigma_x + t_x^2 \Sigma_d + \mathbf{d}\mathbf{d}^T \sigma_{t_x}^2 + \mathbf{e}\mathbf{e}^T \sigma_{r_x}^2 & t_x \Sigma_d \\ t_x \Sigma_d & \Sigma_d + \mathbf{e}\mathbf{e}^T \sigma_{t_z}^2 \end{bmatrix}$$

To obtain the new depth and variance at a point x, we must define an interpolation function for the patch surrounding this point. For a linear interpolant, the equation is

$$d = d_i\frac{(x_{i+1} - x)}{(x_{i+1} - x_i)} + d_{i+1}\frac{(x - x_i)}{(x_{i+1} - x_i)}$$
$$= (1 - \lambda)d_i + \lambda d_{i+1},$$
$$\text{where } \lambda = \frac{(x - x_i)}{(x_{i+1} - x_i)}$$

$$\frac{\partial d}{\partial d_i} = \frac{(x_{i+1} - x)}{(x_{i+1} - x_i)} = (1 - \lambda)$$

$$\frac{\partial d}{\partial x_i} = -\frac{(d_{i+1} - d_i)(x_{i+1} - x_i)}{(x_{i+1} - x_i)^2}$$
$$= -m(1 - \lambda)$$
$$\text{where } m = \frac{(d_{i+1} - d_i)}{(x_{i+1} - x_i)}$$

and the associated Jacobian is

$$\frac{\partial(d)}{\partial(x_i, x_{i+1}, d_i, d_{i+1})}$$
$$= \begin{bmatrix} -m(1 - \lambda) & -m\lambda & (1 - \lambda) & \lambda \end{bmatrix}$$

The variance of the new depth estimate is thus

$$\text{var}(d) = m^2[(1 - \lambda)^2\sigma_{x_i}^2 + \lambda^2\sigma_{x_{i+1}}^2]$$
$$+ (1 - t_x m)^2[(1 - \lambda)^2\sigma_{d_i}^2 + \lambda^2\sigma_{d_{i+1}}^2]$$
$$+ m^2[d^2\sigma_{t_x}^2 + \sigma_{t_z}^2] + \sigma_{t_x}^2$$

Each of the above four terms can be analyzed separately. The first term in the above equation, which involves $\sigma_{x_i}^2$, depends on the positional uncertainty of the points in the old map. It can either be ignored (if each disparity element represents the disparity at its *center*), or $\sigma_{x_i}^2$ can be set to 1/2. The second term is a blend of the variances at the two endpoints of the interpolated interval. Note that for $\lambda = 1/2$, the variance is actually reduced by half (the average of two uncertain measurements is more certain). It may be desirable to use a pure blend $[(1 - \lambda)\sigma_{d_i}^2 + \lambda\sigma_{d_{i+1}}^2]$ to eliminate this bias. The second term also encodes the interaction between the disparity uncertainty and the disparity gradient m. The third term encodes the interaction between the disparity gradient and the camera translation and pan uncertainty. The final term is the uncertainty in camera forward motion, which should in practice be negligible.

References

1. P. Anandan, "Computing dense displacement fields with confidence measures in scenes containing occlusion," *Proc. DARPA Image Understanding Workshop*, pp. 236-246, 1984.
2. N. Ayache and O.D. Faugeras, "Maintaining representations of the environment of a mobile robot," *Proc. 4th Intern. Symp. Robotics Res.* 1987.
3. H.H. Baker, "Multiple-image computer vision," *Proc. 41st Photogrammetric Week*, Stuttgart, West Germany, pp. 7-19, 1987.
4. R.C. Bolles, H.H. Baker, and D.H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *Intern. J. Computer Vision* **1**:7-55, 1987.
5. T.J. Broida and R. Chellappa, "Kinematics and structure of a rigid object from a sequence of noisy images," *Proc. Workshop on Motion: Representation and Analysis*, pp. 95-100, 1986.
6. A.R. Bruss and B.K.P. Horn, "Passive navigation," *Comput. Vision, Graphics, and Image Process.* **21**:3-20, 1983.
7. J. Canny, "A computational approach to edge detection.," *IEEE Trans. PAMI* **8**:679-698, 1986.
8. J.R. Wertz (ed.), *Spacecraft Attitude Determination and Control*. D. Reidel: Dordrecht, 1978.
9. O.D. Faugeras, N. Ayache, B. Faverjon, and F. Lustman, "Building visual maps by combining noisy stereo measurements," *Proc. IEEE Intern. Conf. Robotics and Automation*, San Francisco, California pp. 1433-1438; 1986.
10. J. Hallam, "Resolving observer motion by object tracking," *Proc. 8th Intern. Joint Conf. Artif. Intelli.* Karlsruhe, 1983.
11. D.J. Heeger, "Optical flow from spatiotemporal filters," *Proc. 1st Intern. Conf. Computer Vision*, London, pp. 181-190, 1987.
12. B.K.P. Horn and B.G. Schunck, "Determining optical flow," *Artificial Intelligence*, **17**:185-203, 1981.
13. H.C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proc. Roy. Soc. London* B **208**:385-397, 1980.
14. J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *J. Am. Stat. Assoc.* **82**:76-89, 1987.
15. L.H. Matthies, "Dynamic stereo," Ph.D. thesis, Carnegie Mellon University, 1989.
16. L.H. Matthies and T. Kanade, "The cycle of uncertainty and constraint in robot perception," *Proc. Intern. Symp. Robotics Research*, 1987.
17. L.H. Matthies and S.A. Shafer, "Error modeling in stereo navigation," *IEEE J. Robotics and Automation*, pp. 239-248, 1987.
18. P.S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 2. Academic Press: New York, 1982.
19. P.S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 1. Academic Press, New York, 1979.
20. J.E.W. Mayhew and J.P. Frisby, "Psychophysical and computational studies towards a theory of human stereopsis," *Artificial Intelligence*, **17**:349-408, 1981.
21. E.M. Mikhail, *Observations and Least Squares*. University Press of America: Lanham, MD, 1976.
22. H.-H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. PAMI* **8**:565-593, 1986.
23. V. Nalwa, "On detecting edges," *IEEE Trans. PAMI* **8**:699-714, 1986.

24. Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. PAMI* **7**:139–154, 1985.
25. P. Rives, E. Breuil, and B. Espiau, "Recursive estimation of 3d features using optical flow and camera motion," *Proc. Conf. Intelligent Autonomous Systems*, pp. 522–532, 1986.
26. M.A. Snyder, "Uncertainty analysis of image measurements," *Proc. DARPA Image Understanding Workshop*, Los Angeles, pp. 681–693, 1987.
27. I. Sobel, "On calibrating computer controlled cameras for perceiving 3-d scenes," *Artificial Intelligence* **5**:185–198, 1974.
28. R. Szeliski, *Bayesian modeling of uncertainty in low-level vision*. Ph.D. thesis, Carnegie Mellon University, 1988.
29. R. Szeliski, "Regularization uses fractal priors," *Proc. AAAI-87*, Seattle, pp. 749–754, 1987.
30. Technical Staff, The Analytic Sciences Corporation, *Applied Optimal Estimation*, MIT Press: Cambridge, MA, 1974.
31. D. Terzopoulos, "Image analysis using multigrid relaxation methods," *IEEE Trans. PAMI* **8**:129–139, 1986.
32. D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. PAMI* **8**:413–424, 1986.
33. A.M. Waxman and J.J. Duncan, "Binocular image flows," *Proc. Workshop on Motion: Representation and Analysis*, Kiawah Island, SC, pp. 31–38, 1986.
34. G. Xu, S. Tsuji, and M. Asada, "Coarse-to-fine control strategy for matching motion stereo pairs," *Proc. IJCAI*, Los Angeles, pp. 892–894, 1985.