An Estimation-Theoretic Framework for Image-Flow Computation

Ajit Singh

Vision and Robotics Group
Department of Computer Science
Columbia University, New York.

Advanced Imaging Group
Siemens Corporate Research
Princeton, New Jersey

CUCS-510-89

Abstract

Image-flow is a major source of three-dimensional information. This paper describes a new framework for computing image-flow from time-varying imagery. In this framework, image-flow information is classified into two categories - conservation information and neighborhood information. Each type of information is recovered in the form of an estimate accompanied by a covariance-matrix. Image-flow is then computed by fusing the two estimates using estimation-theoretic techniques. This framework offers the following principal advantages. Firstly, it allows estimation of certain types of discontinuous flow-fields without any a-priori knowledge about the location of discontinuities. The flow-fields thus recovered are not blurred at motion-boundaries. Secondly, covariance matrices (or alternatively, confidence-measures) are associated with the estimate of image-flow at each stage of computation. The estimation-theoretic nature of the framework and its ability to provide covariance matrices make it very useful in the context of applications such as incremental estimation of scene-depth using techniques based on Kalman filtering. In this paper, an algorithm based on this framework is used to recover image-flow from two image-sequences. To illustrate an application, the image-flow estimates and their covariance matrices thus obtained are also used to recover scene-depth.

Address for correspondence

Ajit Singh
Siemens Corporate Research, Inc.
755 College Road East
Princeton, NJ 08540

An Estimation-Theoretic Framework for Image-Flow Computation

Abstract

Image-flow is a major source of three-dimensional information. This paper describes a new framework for computing image-flow from time-varying imagery. In this framework, image-flow information is classified into two categories - conservation information and neighborhood information. Each type of information is recovered in the form of an estimate accompanied by a covariance-matrix. Image-flow is then computed by fusing the two estimates using estimation-theoretic techniques. This framework offers the following principal advantages. Firstly, it allows estimation of certain types of discontinuous flow-fields without any a-priori knowledge about the location of discontinuities. The flow-fields thus recovered are not blurred at motion-boundaries. Secondly, covariance matrices (or alternatively, confidence-measures) are associated with the estimate of image-flow at each stage of computation. The estimation-theoretic nature of the framework and its ability to provide covariance matrices make it very useful in the context of applications such as incremental estimation of scene-depth using techniques based on Kalman filtering. In this paper, an algorithm based on this framework is used to recover image-flow from two image-sequences. To illustrate an application. the image-flow estimates and their covariance matrices thus obtained are also used to recover scene-depth.

1 Introduction

Image-flow is a commonly used representation for visual-motion. It assigns to each point on the visual-field, a two-dimensional velocity vector that depicts the projection of the instantaneous three-dimensional velocity of the corresponding point in the scene. Typically, all the information that is available about a dynamic scene is an image-sequence. The image-flow field must be computed from the image-sequence. Furthermore, the process of image-flow computation must make use of *local* spatial and temporal neighborhoods. This restriction is generally imposed for reasons of computational efficiency as well as physiological plausibility.

This paper describes a new estimation-theoretic framework for image-flow computation. The principal advantages offered by this framework are as follows. (i) Covariance matrices (or alternatively, confidence-measures) are associated with the estimate of image-flow at each stage of computation. (ii) It is possible to estimate certain types of discontinuous flow-fields without any a-priori knowledge about the location of discontinuities. The flow-fields thus recovered are not blurred at motion-boundaries. (iii) Because of its estimation-theoretic nature, the framework lends itself naturally to incremental estimation of scene-depth from image-flow using techniques based on Kalman filtering. A contribution of this framework that is not discussed in this paper because of space limitations is that it serves to unify a very wide class of existing techniques for image-flow computation. The issue of unification will be discussed in a sequel paper. Before giving an overview of this framework, a brief review of the state of the art will be in order.

It is well understood [4, 15] that by using local measurements alone, the true velocity can be recovered only in those image regions that have sufficient local intensity variation, such as intensity corners, textured-regions. etc. This constitutes the well known aperture problem. Velocity must be propagated from regions of full information. such as corners etc., to regions of partial or no information. This implies that any approach to local computation of image-flow must incorporate *two* functional steps. In the first step, local information about velocity is recovered using the image-intensity distribution in small spatiotemporal neighborhoods. In the second step, the local information is propagated into neighboring regions to recover the correct image-flow. The past research is summarized below in light of these two steps. A detailed review can be seen in [2. 19]. Most of the current frameworks for image-flow computation use one of the following three basic approaches for the first step mentioned above: (i) correlation-based approach [4. 18], (ii) gradient-based approach [7, 8, 11, 15, 22] and (iii) spatiotemporal energy based approach [1, 9]. The output of the first step is in the form of initial-estimates that are updated iteratively in the second step. For the second step, the current frameworks use either a smoothness constraint [4. 10. 11, 15] or the analytical structure of image-flow [14, 21].

In the framework described here, the image-flow information available in time-varying imagery is classified into two categories - *conservation information* and *neighborhood information*. In terms of the two-step solution suggested above. conservation information is extracted in the first-step. I call it conservation information because it is derived from the imagery by using the assumption of conservation of some image-property over time. Typically, this property is intensity [8. 11, 15], some spatiotemporal derivative of intensity [7] or intensity distribution in a small spatial neighborhood [4, 18] etc. Other choices are possible. e.g., color. Similarly, neighborhood information corresponds to the second step. I call it neighborhood information because it is derived by using the knowledge of velocity distribution in small spatial neighborhoods in the visual-field. Each type of information is recovered in the form of an estimate accompanied by a covariance-matrix. Image-flow is then computed by fusing the two estimates on the basis of their covariance-matrices.

The organization of this paper is as follows. In section 2. I show how to recover conservation information. For simplicity of presentation, I use a correlation-based approach. In the sequel paper. I show that one could use any one of the three basic approaches to recover conservation information. In section 3. I discuss the procedure for recovering neighborhood information. I also show that image-flow computation can be posed as a problem of combining conservation information and neighborhood information optimally (in a statistical sense). I present an iterative solution to this problem. I show an algorithm based on this framework in section 4 and describe the results of applying this algorithm to a variety of image sequences in section 5. In order to put this framework in context of an application, I also show the results of using the image-flow estimates to recover scene-depth using a variant of the Kalman filtering-based technique proposed by Matthies et. al [13]. Finally, I give concluding remarks in section 6.

2 Step 1: Conservation information

An implicit assumption on which most image-flow computation techniques are based is that some image-property is conserved over time. In other words, in each image of a sequence, the projection of a given moving point in the scene will have the same value of the conserved property. Factors that affect the robustness of the choice of conserved property are illumination, type of motion (rotational/translational), noise and digitization effects etc. [4, 19]. For reasons of computational simplicity, I use the Laplacian of intensity (computed by the difference-of-Gaussians operation using the masks suggested by Burt [6].) as the conserved property. I refer to the Laplacian image as just "image" for sake of brevity.

Based on the assumption of conservation, estimating image-flow using a correlation-based approach [4] amounts to an explicit search for the best match for a given pixel of an image in a search-area in subsequent images of the sequence. The extent of the search-area can be decided on the basis of a-priori knowledge about the maximum possible displacement between two images or by using a hierarchical strategy [4]. Correlation gives a *response*, i.e., a matching-strength, at each pixel in the search area. Thus, the search area can be visualized as covered with a "response-distribution". Anandan [4] had shown that using the sum-of-squared-differences (SSD) offers several computational advantages over correlation. Using SSD, which is a measure of mismatch, one obtains an "error-distribution" over the search area. The procedure for obtaining error distribution and converting it into response-distribution is discussed below.

For each pixel $\mathcal{P}(x,y)$ at location (x,y) in the first image $\mathcal{I}_1$, a correlation-window $\mathcal{W}_p$ of size $(2n+1) \times (2n+1)$ is formed around the pixel. A search-window $\mathcal{W}_s$ of size $(2N+1) \times (2N+1)$ is established around the pixel at location (x,y) in the second image $\mathcal{I}_2$. The $(2N+1) \times (2N+1)$ sample of error-distribution is computed using sum-of-squared-differences as:

$$\mathcal{E}_c(u,v) = \sum_{i=-n}^{n}\sum_{j=-n}^{n}(\mathcal{I}_1(x+i,y+j) - \mathcal{I}_2(x+u+i,y+v+j))^2$$
$$-N \leq u,v \leq +N \tag{1}$$

The $(2N+1)\times(2N+1)$ sample of response-distribution is computed as follows:

$$\mathcal{R}_c(u,v) = e^{-k\mathcal{E}_c(u,v)}$$
$$-N \leq u,v \leq +N \tag{2}$$

The choice of an exponential function for converting error-distribution into response-distribution is based primarily on computational reasons. Firstly, it is well behaved when error approaches zero. Secondly, the response obtained with an exponential function varies continuously between zero and unity over the entire range of error.

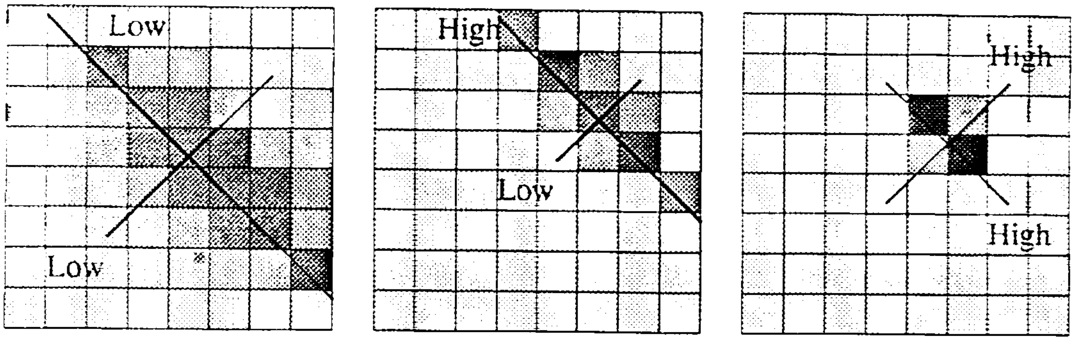
I suggest that response-distribution be interpreted as follows. Each point in the search area is a candidate for the "true match". However, a point with a small response is less likely to be the true match, as compared to a point with a high response. Assuming that the time elapsed between two successive images is unity, each point in the search area represents a point in $u-v$ space. In estimation-theoretic terms, each of these points can be thought of as a *measurement* of the true velocity. Further, the response at the point can be thought of as a weight that reflects our faith in the measurement. One could compute an *estimate* of velocity using, for instance, a weighted-least-squares approach. Under the assumption of additive and zero-mean errors, one could also associate a covariance-matrix with this estimate. Quantitatively, the weighted-least-squares based estimate, denoted by $U_{cc} = (u_{cc}, v_{cc})$, is given by:

$$u_{cc} = \frac{\sum_u\sum_v \mathcal{R}_c(u,v)u}{\sum_u\sum_v \mathcal{R}_c(u,v)}$$
$$v_{cc} = \frac{\sum_u\sum_v \mathcal{R}_c(u,v)v}{\sum_u\sum_v \mathcal{R}_c(u,v)} \tag{3}$$

and the covariance-matrix associated with this estimate is given by:

$$\mathcal{S}_{cc} = \begin{pmatrix} \frac{\sum_u\sum_v \mathcal{R}_c(u,v)(u-u_{cc})^2}{\sum_u\sum_v \mathcal{R}_c(u,v)} & \frac{\sum_u\sum_v \mathcal{R}_c(u,v)(u-u_{cc})(v-v_{cc})}{\sum_u\sum_v \mathcal{R}_c(u,v)} \\ \frac{\sum_u\sum_v \mathcal{R}_c(u,v)(u-u_{cc})(v-v_{cc})}{\sum_u\sum_v \mathcal{R}_c(u,v)} & \frac{\sum_u\sum_v \mathcal{R}_c(u,v)(v-v_{cc})^2}{\sum_u\sum_v \mathcal{R}_c(u,v)} \end{pmatrix} \tag{4}$$

where the summation is carried out over $-N \leq u,v \leq +N$. It is known [5] that reciprocals of the eigenvalues of the covariance-matrix serve as confidence-measures associated with the estimate, along the directions given by

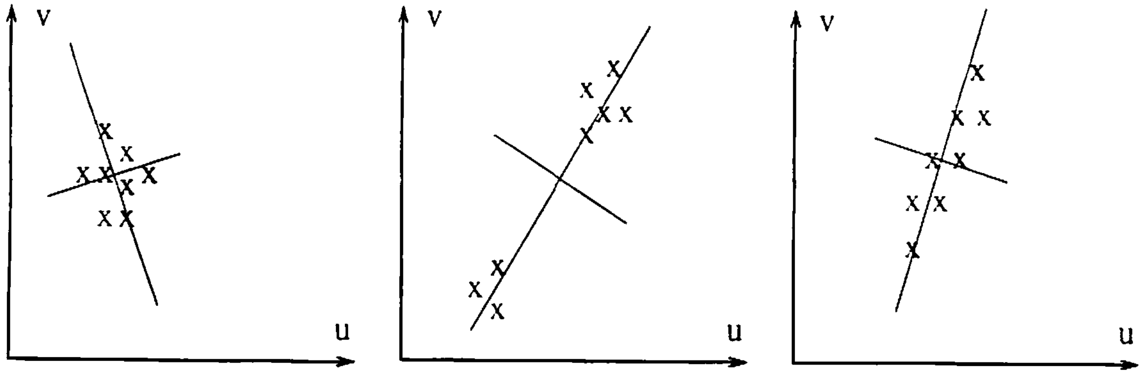


(a) In a uniform region (b) Near an edge (c) Near a corner

Figure 1: Response-distribution over the search-window for some representative examples - darker the pixel, higher the response. The labels "high" and "low" refer to the confidence measures associated with the eigenvectors.

the corresponding eigenvectors. Figure 1 shows the eigenvectors and the corresponding confidence measures for some typical response distributions. Further, these eigenvectors correspond to the principal axes of response distribution. Principal axes have been used to represent velocity earlier by Scott [18].

Summarizing, there are three essential steps underlying the computation of conservation information. They are: (i) selecting the conserved quantity and deriving it from intensity imagery, (ii) computing error-distribution and response-distribution over the search-area in the velocity-space and (iii) interpreting response distribution, i.e., computing an estimate of velocity along with a covariance-matrix. The estimate, U_{cc}, can be thought of as the "initial estimate" that serves as input (along with the covariance $\mathcal{S}_{cc}$) to the velocity propagation procedure. As mentioned earlier, velocity propagation is accomplished using neighborhood information. Before discussing neighborhood information, the following clarification would be in order. In interpreting the response-distribution, I have assumed that it is unimodal. This assumption does get violated in the presence of texture, specially if the size of the search-window is greater than the scale of intensity variations. The weighted-least-squares approach used above "averages out" the various peaks, giving an incorrect estimate of velocity. However, since the "spread" of the distribution is large in this case (as compared to the situation where the response-distribution has a single well defined peak), the confidence associated with the estimate will be low. In essence, although the procedure for interpreting the response-distribution gives an incorrect estimate if the distribution is not unimodal, it does associate a low confidence with the (incorrect) estimate. Further, the problem of multiple peaks can be alleviated, at least partly, by using three images to compute conservation information. This is done by computing two response-distributions - one between the current image and the previous image and other between the current image and the next image - and adding the two



(a) Uniform region (b) Region boundary (c) Gradual depth change

Figure 2: Velocity distribution for some representative neighborhoods.

appropriately.

3 Step 2: Neighborhood information

The objective of the second step in image-flow recovery is to propagate velocity by using neighborhood information. Assume for a moment that the velocity of each pixel in a small neighborhood around the pixel under consideration is known. One could plot these velocities as points in $u - v$ space giving a neighborhood velocity distribution. Some typical distributions are shown in figure 2. What can one say about the velocity of the central pixel (which is unknown)? Barring the case where the central pixel lies in the vicinity of a motion-boundary, it is reasonable to assume that it is "similar" to velocities of the neighboring pixels. In statistical terms, the velocity of each point in the neighborhood can be thought of as a *measurement* of the velocity of the central pixel. It is reasonable to assume that all of these measurements are not equally reliable - they must be weighted differently if used to compute an estimate of velocity of the central pixel. I weight the velocities of various pixels in the neighborhood according to their distance from the central pixel - larger the distance, smaller the weight. Specifically, I use a Gaussian mask. Based on this information, a weighted-least-squares estimate of velocity, $\overline{U}$, can be computed. Further, assuming additive and zero mean errors, a covariance-matrix, $\mathcal{S}_n$ can be associated with this estimate. The estimate and the covariance-matrix thus obtained serve as the "opinion" of the neighborhood regarding the velocity of the central pixel (as opposed to those obtained from conservation information that reflect the central pixel's own opinion).

Quantitatively, if the neighborhood size is $(2w+1) \times (2w+1)$, the velocities of these $(2w+1)^2$ pixels map to the points (u_i, v_i) in $u - v$ space (where $1 \leq i \leq (2w+1)^2$) and the weight assigned to the point (u_i, v_i) is

$\mathcal{R}_n(u_i, v_i)$, the weighted-least-squares based estimate $\overline{U} = (\overline{u}, \overline{v})$, of velocity of the central pixel is given by:

$$\begin{aligned} \overline{u} &= \frac{\sum_u \sum_v \mathcal{R}_n(u,v)u}{\sum_u \sum_v \mathcal{R}_n(u,v)} \\ \overline{v} &= \frac{\sum_u \sum_v \mathcal{R}_n(u,v)v}{\sum_u \sum_v \mathcal{R}_n(u,v)} \end{aligned} \tag{5}$$

and the covariance-matrix associated with this estimate is given by:

$$\mathcal{S}_n = \begin{pmatrix} \frac{\sum_i \mathcal{R}_n(u_i,v_i)(u_i-\overline{u})^2}{\sum_i \mathcal{R}_n(u_i,v_i)} & \frac{\sum_i \mathcal{R}_n(u_i,v_i)(u_i-\overline{u})(v_i-\overline{v})}{\sum_i \mathcal{R}_n(u_i,v_i)} \\ \frac{\sum_i \mathcal{R}_n(u_i,v_i)(u_i-\overline{u})(v_i-\overline{v})}{\sum_i \mathcal{R}_n(u_i,v_i)} & \frac{\sum_i \mathcal{R}_n(u_i,v_i)(v_i-\overline{v})^2}{\sum_i \mathcal{R}_n(u_i,v_i)} \end{pmatrix} \tag{6}$$

where the summation is carried out over $1 \leq i \leq (2w+1)^2$.

At this point, we have two estimates of velocity, U_{cc} and $\overline{U}$ - from conservation and neighborhood information respectively, each with a covariance-matrix. An estimate of velocity that takes both conservation information and neighborhood information into account can now be computed as follows. Since this estimate is a point in $u - v$ space, its distance from $\overline{U}$, weighted appropriately by the corresponding covariance matrix, represents the error in satisfying neighborhood information. I refer to this error as *neighborhood error*. Similarly, the distance of this point from U_{cc}, weighted appropriately, represents the error in satisfying conservation information. I refer to this error as *conservation error*. Computing the velocity estimate, therefore, amounts to finding a point in $u - v$ space that minimizes the sum of neighborhood error and conservation error.

In quantitative terms, neighborhood error is a quadratic form commonly used in estimation theory [5] and is given by:

$$(U - \overline{U})^T \mathcal{S}_n^{-1} (U - \overline{U}) \tag{7}$$

Similarly, conservation error is the following quadratic form:

$$(U - U_{cc})^T \mathcal{S}_{cc}^{-1} (U - U_{cc}) \tag{8}$$

and the sum of conservation error and neighborhood error represents the squared error in the velocity estimate U. Statistically speaking, the optimal estimate of velocity is the one that minimizes the mean squared error over the visual field. That is:

$$\int\int \left[(U - \overline{U})^T \mathcal{S}_n^{-1} (U - \overline{U}) + (U - U_{cc})^T \mathcal{S}_{cc}^{-1} (U - U_{cc})\right] dx dy = MINIMUM \tag{9}$$

Calculus of variations can be used to derive the optimal estimate. Let ∇_U be defined as follows:

$$\nabla_U = \begin{pmatrix} \frac{\partial}{\partial u} \\ \frac{\partial}{\partial v} \end{pmatrix} \tag{10}$$

The condition for minimum mean squared error can be written as:

$$\nabla_U \left[\int \int \left[(U - U_{cc})^T \mathcal{S}_{cc}^{-1} (U - U_{cc}) + (U - \overline{U})^T \mathcal{S}_n^{-1} (U - \overline{U}) \right] dx dy \right] = 0 \tag{11}$$

which gives [5]:

$$\mathcal{S}_{cc}^{-1} (U - U_{cc}) + \mathcal{S}_n^{-1} (U - \overline{U}) = 0 \tag{12}$$

In this equation, U_{cc} and $\mathcal{S}_{cc}$ are derived directly from the underlying intensity pattern in the image. Therefore, they are known (and fixed) for a each pixel. $\overline{U}$ and $\mathcal{S}_n$, on the other hand, are derived on the assumption that velocity of each pixel in the neighborhood is known in advance from an independent source. This assumption is invalid in practice. Hence, $\overline{U}$ and $\mathcal{S}_n$ are unknown and the velocity U cannot be derived directly from equation 12. However, equation 12 is available at all the pixels in any given neighborhood in the image. If the conditions discussed below are satisfied, we essentially have a system of coupled linear equations that can be solved by an iterative technique such as Gauss-Siedel relaxation algorithm [16]. The iterative solution can be written as [16]:

$$\begin{aligned} U^{k+1} &= \left[\mathcal{S}_{cc}^{-1} + \mathcal{S}_n^{-1} \right]^{-1} \left[\mathcal{S}_{cc}^{-1} U_{cc} + \mathcal{S}_n^{-1} \overline{U}^k \right] \\ U^0 &= U_{cc} \end{aligned} \tag{13}$$

and the covariance matrix associated with the final estimate of velocity is given by $[\mathcal{S}_{cc}^{-1} + \mathcal{S}_n^{-1}]^{-1}$, where $\mathcal{S}_n^{-1}$ is computed from the final iteration. The eigenvalues of this matrix depict the confidence measures corresponding to the final estimate. The notion of final (post-propagation) covariance matrix is novel and unique to this framework. It serves several purposes. Qualitatively, it indicates what regions in the image have the most reliable image-flow estimates from the viewpoint of applicability to high-level interpretation. Quantitatively, it serves as an essential input to procedures for incremental scene-depth computation that use estimation theoretic techniques such as Kalman filtering. This is discussed in appendix A and used in depth-estimation experiments reported in the next section.

The two conditions that must be satisfied for the iterative solution to converge are discussed below. Firstly, for equation 12 to represent a system of coupled *linear* equations, $\mathcal{S}_n$ must be a constant and must be known in advance. Such is not the case here. In the current implementation, I obtain $\mathcal{S}_n$ from the neighborhood velocity distribution corresponding to the previous iteration. However, I have found empirically that either of the eigenvalues of $\mathcal{S}_n$ does not change by more than about 15% from the beginning to the end of the iterative procedure. This holds true particularly for the pixels that do not lie on a motion boundary. Secondly, for the iterative procedure to converge irrespective of the value of initial estimate U^0, both S_{cc}^{-1} and S_n^{-1} must be positive definite. As discussed in [20], this criterion is generally satisfied in real imagery except in pathological cases such as absolutely flat regions.

So far, I have assumed that the pixel under consideration does not lie on a motion-boundary and that neighborhood velocity distribution forms a single cluster in $u - v$ space. In the following discussion, I will analyze the performance of the framework at motion-boundaries. Specifically, I will show that (i) the procedure discussed above for using neighborhood information is still justified and (ii) in absence of texture, it does a better job of preserving the step-discontinuities in the flow-field as compared to conventional smoothing-based procedures. For this purpose, recall that each of the two estimates U_{cc} and $\overline{U}$ maps to a point in $u - v$ space. Similarly, each of the two covariance matrices S_{cc} and S_n maps to an ellipse that has its center at the respective estimate and that has its major and minor axes equal to the eigenvalues of the covariance-matrix. Therefore, each iteration amounts to finding a point in $u - v$ space that has the minimum weighted sum of squared perpendicular distances from the axes of the two ellipses - the eigenvalues serving as weights.

The behavior of this procedure in the vicinity of a motion-boundary is depicted in figure 3a. For the conservation-ellipse E_{cc}, only the major axis is shown because the minor axis will be very small in this region. In other words, all that conservation information tells (with high confidence) about the velocity of the central pixel is that it lies somewhere along the major axis of the ellipse E_{cc}. Velocities of neighboring points are also plotted from the previous iteration. Given that there is no texture in vicinity of the boundary (i.e., conservation information is reliable) and that the boundary corresponds to a step-discontinuity in the flow-field, the velocities of neighboring points form two clusters in $u - v$ space. As a result, the minor axis of the neighborhood-ellipse E_n will be very small. In other words, all that neighborhood information tells (with high confidence) about the velocity of the central pixel is that it lies somewhere along the major axis of the ellipse E_n. Since the "correct" velocity will lie in one of the two clusters, this opinion of neighborhood is correct. In other words, the iterative update procedure developed for the non-boundary pixels is justified even for pixels that lie on a motion-boundary. Furthermore, since the velocities of the neighboring pixels are derived from conservation information (at the beginning of the iterative procedure), one of the two clusters will be very close to the conservation-constraint for the central pixel. This is depicted in figure 3a. As a result, the updated velocity for the central pixel, which is given by the intersection of the two major axes (if the minor axes are also used, the updated velocity will be slightly offset), will be very close to one of the

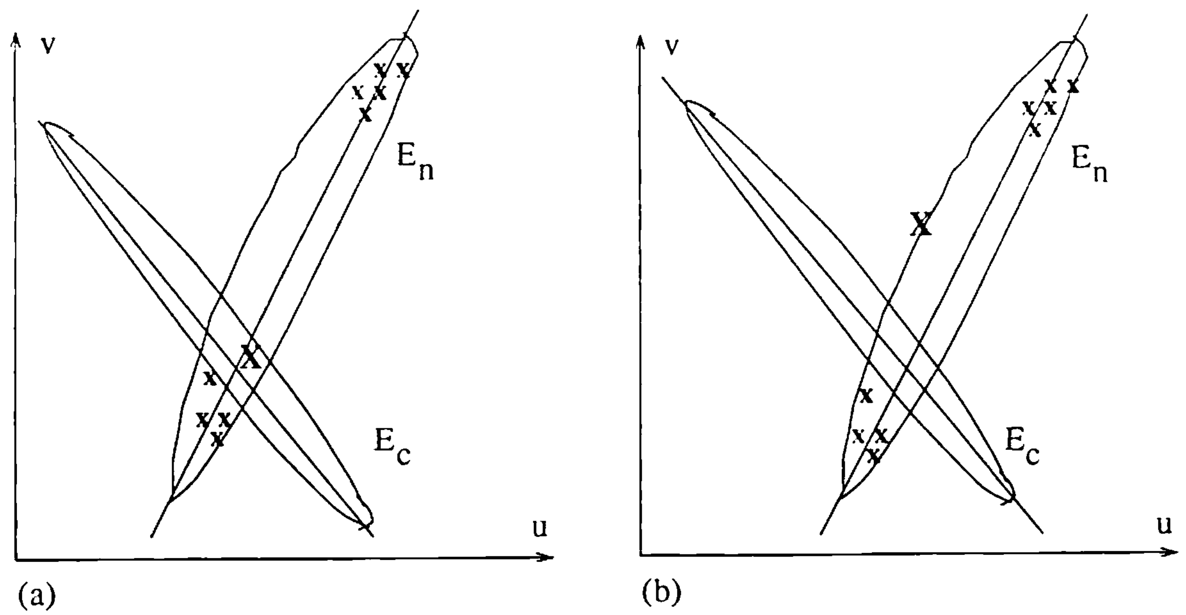


Figure 3: Performance at motion-boundaries.

clusters. This cluster corresponds to that side of motion boundary (in the image) with which the velocity of the central pixel is more consistent. Effectively, the pixel under consideration is *binned* to the correct side of the motion boundary. For purpose of comparison, the result of conventional smoothing [4, 11] is shown in figure 3b. Clearly, the updated velocity lies somewhere in the middle of the two clusters, effectively blurring the flow-field at the boundary.

4 An Algorithm and its implementation

An algorithm based on the new framework is given below followed by the details of its implementation. The algorithm uses three images as its input. It recovers conservation information only once at the onset (steps 1 through 3) and neighborhood information once for each iteration (steps 4 through 6).

Algorithm:

(1) Convolve each image with a Laplacian. (2) Form a $(2n+1) \times (2n+1)$ correlation-window around the pixel in consideration in the central image. Also, form a $(2N+1) \times (2N+1)$ search-window in around the corresponding location in the other two images. Compute the error-distributions over the two search windows and transform them to the corresponding response-distributions, $\mathcal{R}_c^{-1}$ and $\mathcal{R}_c^{+1}$ respectively. Finally, rotate $\mathcal{R}_c^{-1}$ about both vertical and horizontal axes and add it to $\mathcal{R}_c^{+1}$ to compute the resultant response distribution $\mathcal{R}_c$. (3) Compute the estimate U_{cc} and the covariance-matrix S_{cc} from response-distribution using equations 3

and 4 respectively. (4) Form a $(2w+1) \times (2w+1)$ window around the pixel in consideration. Denote each pixel by a distinct index i, where $1 \leq i \leq (2w+1)^2$. Denote the current estimate of the velocity of i^{th} pixel by (u_i, v_i). (For the first iteration, the velocity U_{cc} computed in step 3 can be used as the current estimate). Assign weights $\mathcal{R}_n(u_i, v_i)$ to these velocities. Compute the mean $\overline{U}$ and the covariance-matrix $\mathcal{S}_n$ using equations 5 and 6 and respectively. (5) Update the velocity at the pixel under consideration using equation 13. (6) Repeat steps 4 and 5 until the change in velocity over two successive iterations is less than a threshold. (7) Compute the confidence measures associated with the final estimate of velocity as the eigenvalues of the matrix given by $\mathcal{S}_{cc}^{-1} + \mathcal{S}_n^{-1}$. These confidence measures are associated with the directions of maximum and minimum confidence, i.e., along the eigenvectors.

Implementation Details:

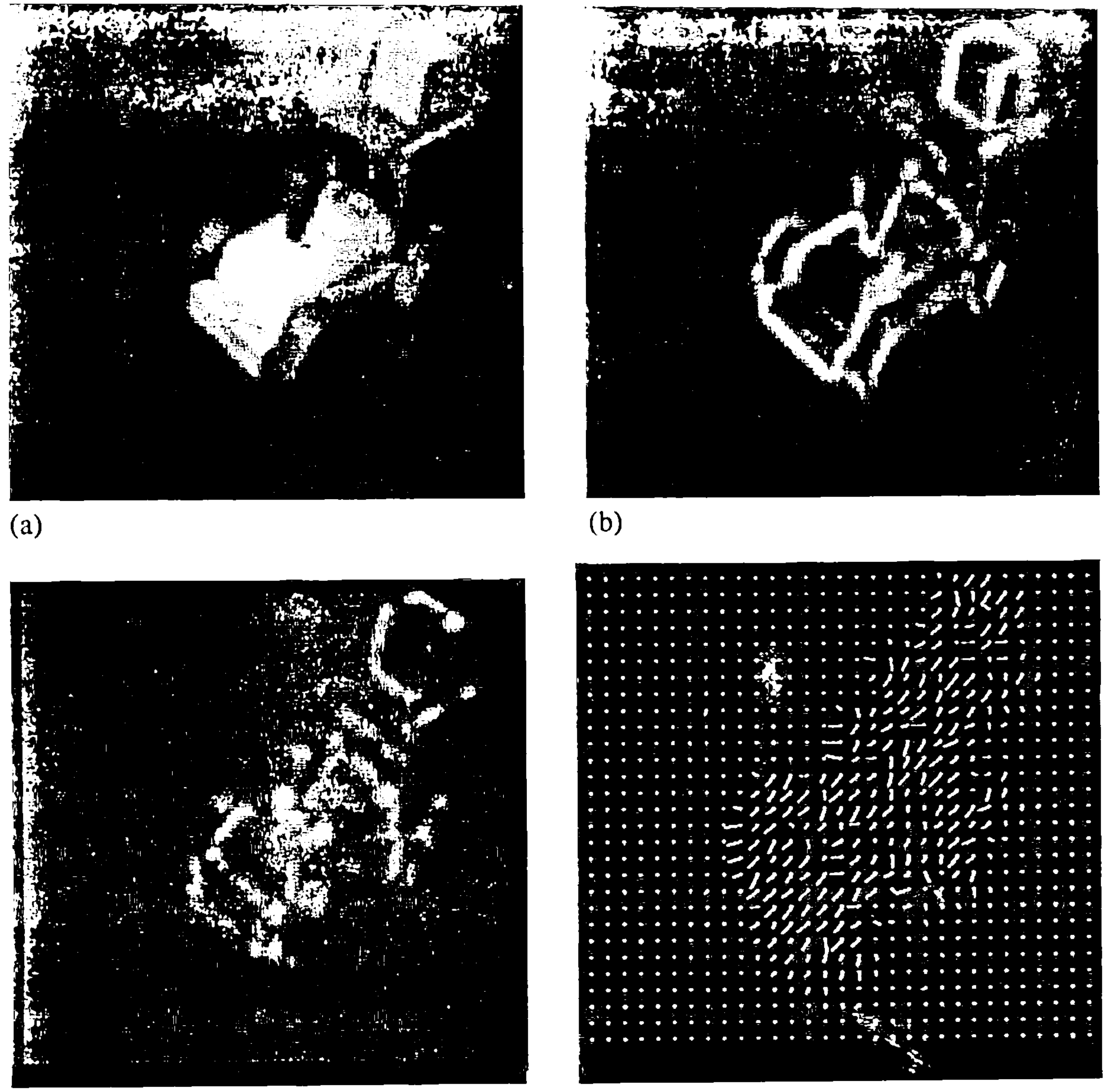
Firstly, one has to establish the parameters N, n, w and k in order to compute response-distribution. The choice of N depends on the maximum possible displacement of a pixel between two frames. If the displacement is small (of the order of one to two pixels per frame), $N = 2$ (i.e., a 5×5 search window) is appropriate. If the displacement is large, one can still use $N = 2$ along with a hierarchical search strategy [4]. The values of n and w are decided on the basis of how many neighbors should contribute their opinion in estimation of velocity of the point under consideration. Too small a neighborhood leads to noisy estimates. Too large a neighborhood tends to smooth out the estimates. Empirically, $n.w = 1$ (i.e., a 3×3 window) appears appropriate. The parameter k is essentially a normalization factor. In the implementation used here, k is chosen in in such a way that the maximum response in the search-window is a fixed number (close to unity). Secondly, inversion of various matrices poses problems when one or more of the eigenvalues are zero or very small. For this reason, singular value decomposition is used for matrix-inversion. Thirdly, the choice of U_{cc} as the starting velocity for the iterative procedure is justified because it denotes the estimate that can be derived from conservation information alone. This ties well with the two-step approach to image-flow recovery - the output of the first step, U_{cc}, serves as an input to the second step. Finally, some criteria has to be established to stop the iterative update process. In the experiments reported in this paper, iteration is stopped when the magnitude of each component of velocity, when rounded to the second decimal place, does not change anywhere in the image.

5 Experiments

The experiments described in this section can be divided into two categories - qualitative and quantitative. For sake of brevity, only one experiment from each category is described. A detailed description of the objectives, methodology and results of each category of experiments is given below.

Qualitative experiments: The objective of this category is to judge the qualitative correctness of flow-fields recovered by the algorithm, specially in terms of preservation of motion-boundaries. The experiment described here uses a toy truck on a flat (and mostly dark) table. Three images are shot as the truck rolls forward. The motion is largely translational, except for in the vicinity of the wheels where it has a small rotational component. Furthermore, the motion-boundaries are expected to show up primarily as step-discontinuities in the flow-field. The images are 256 × 242 in resolution and the maximum image-motion is about three pixels per frame. For image-flow computations, the images are low-pass filtered and subsampled to get a resolution of 128 × 121. At this level of resolution, the maximum image-flow is expected to be between 1 and 1.5 pixels per frame. In the various flow-field images that follow, the velocity vector for only every fourth pixel (in both horizontal and vertical directions) is shown for sake of clarity. Further, the magnitude of velocity is multiplied by a scale-factor of four in order to make the velocity vector clearly visible.

Figures 4 through 7 show various flow-fields and confidence measures. Figure 4a shows the central frame of the original sequence. Figures 4b and 4c show the two confidence measures associated with conservation information at each point in the visual-field. It is clear that the one of the confidence measures is high both at edges and corners of the intensity image whereas the other one is high only at corners. Figure 4d shows the initial estimate of the flow-field (i.e., the velocity U_{cc}). Figure 5 shows the flow-field after iterative velocity propagation (10 iterations), superimposed on the wire-frame of the truck. For sake of comparison, figure 6 shows the flow-field after 10 iterations of conventional smoothing [4, 11] (with the smoothing factor α set to 0.5), also superimposed on the wire-frame. For this purpose, the conservation-based estimate U_{cc} is fed into the smoothing procedure in the manner shown by Anandan [4]. A comparison of figure 5 and 6 clearly shows that the new propagation procedure does an excellent job of preserving motion boundaries. It is apparent that there is very little "bleeding" of velocity from the truck into the background in figure 5. On the other hand, there is considerable blurring of motion-boundaries in figure 6. Figures 7a and 7b show the two confidence measures after propagation. As expected, the confidence has propagated outwards from the pre-propagation high-confidence regions.



(a)

(b)

(c)

(d)

Figure 4: The toy-truck experiment: (a) central frame of the image-sequence. (b),(c) confidence measures associated with conservation information, i.e., the reciprocals of the eigenvalues of the covariance matrix S_{cc} and (d) initial estimate of velocity, i.e., U_{cc}.

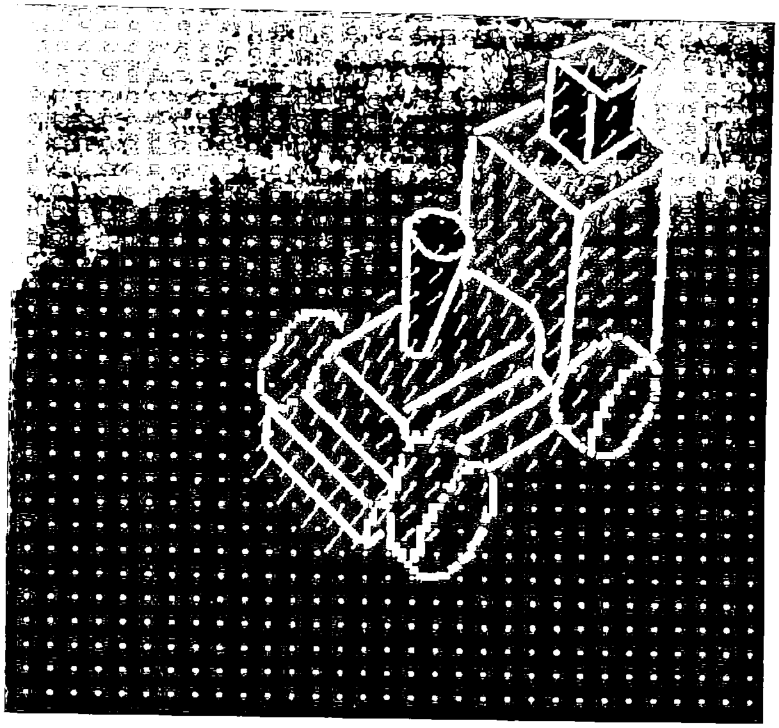


Figure 5: The toy-truck experiment: flow-field after velocity propagation, superimposed on the wire-frame of the truck.

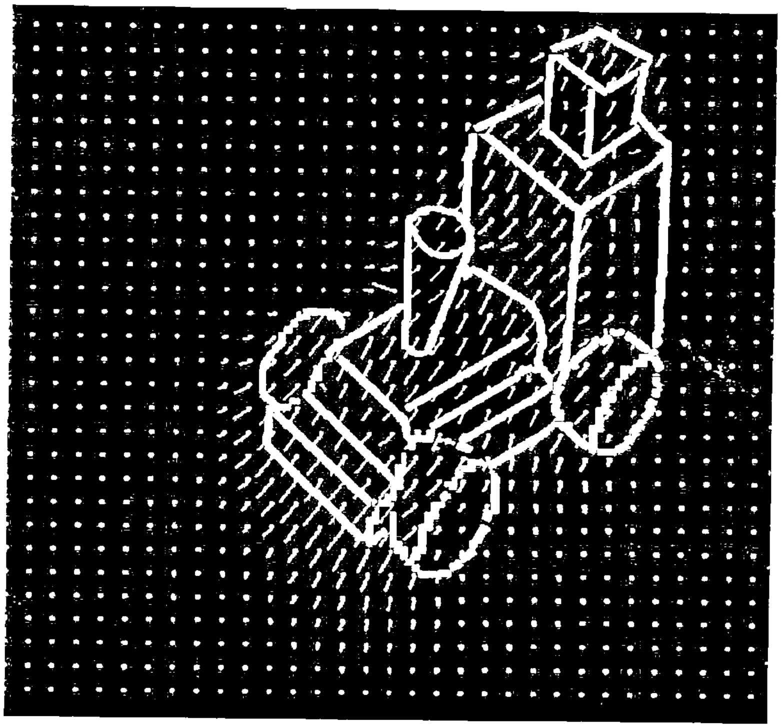


Figure 6: The toy-truck experiment: flow-field after 10 iterations of conventional smoothing, superimposed on the wire-frame of the truck.

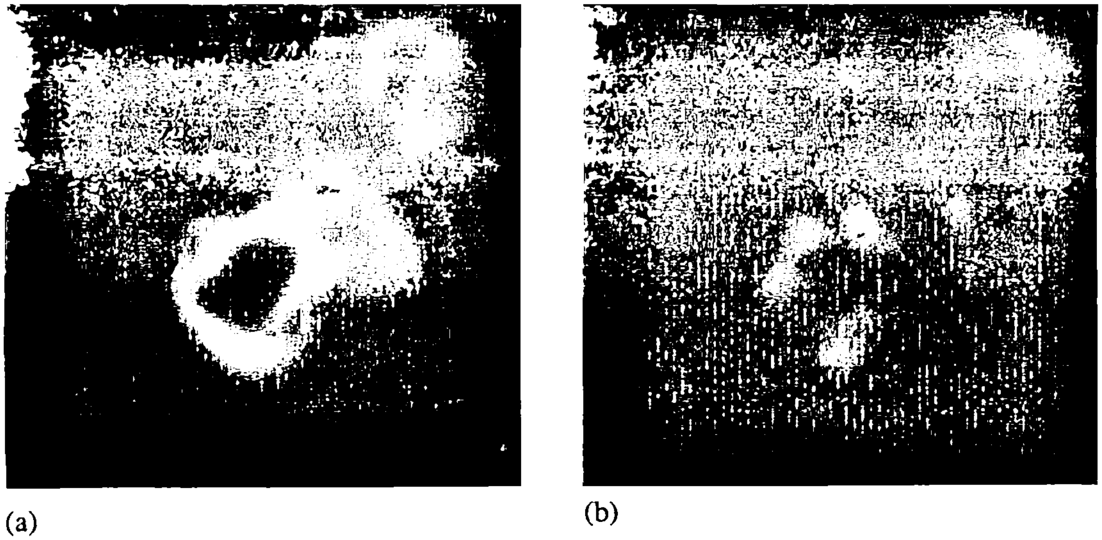


Figure 7: The toy-truck experiment: (a) and (b) confidence measures associated with the flow-field after velocity propagation.

The estimation-theoretic nature of the framework and its ability to provide covariance matrices make it very useful in the context of applications such as incremental estimation of scene-depth using techniques based on Kalman filtering. One such technique was shown by Matthies, Szeliski and Kanade [13]. A variant of their scheme that uses the image-flow estimates and the covariance matrices produced by the new framework is briefly described in appendix A and is used below to recover scene-depth. For this purpose, the toy-truck experiment is repeated with the truck stationary, the camera looking from top (about 15 inches above the truck) and undergoing a one-dimensional translation in a plane perpendicular to its optical axis. Eleven frames are shot at regular intervals as the camera translates horizontally by 1.5 inches. The true depth-map (obtained with a laser range-finder) is shown in figure 8. The depth-map obtained after eleven frames is plotted in figure 9. It is apparent that the depth-estimates are very good. For sake of comparison, the depth-map obtained after eleven frames using the image-flow estimates obtained from the smoothing-based implementation described earlier is plotted in figure 10. It is apparent that the blurring of depth-discontinuities is much more prominent in figure 10. It must be emphasized that the objective of this exercise (of depth-estimation) is to put the new-framework in the context of an application, rather than to make any claims about the performance of a specific depth-estimation scheme.

Quantitative experiments: The general objective of this category of experiments is to judge the quantitative correctness of the flow-fields. In order to accomplish this, the "ground-truth" flow-field must be known.

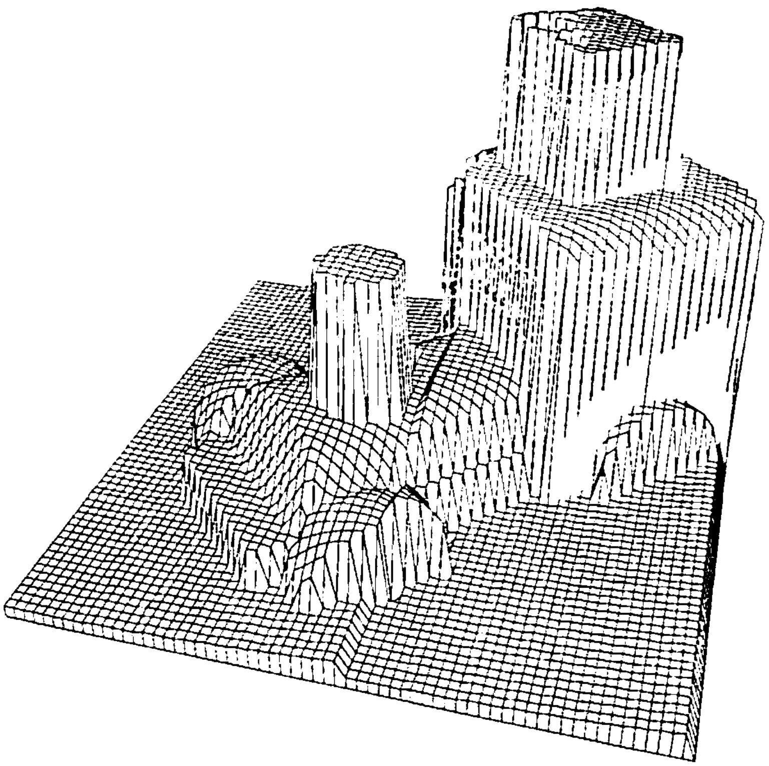


Figure 8: The toy-truck experiment: the true depth-map obtained with a laser range-finder.

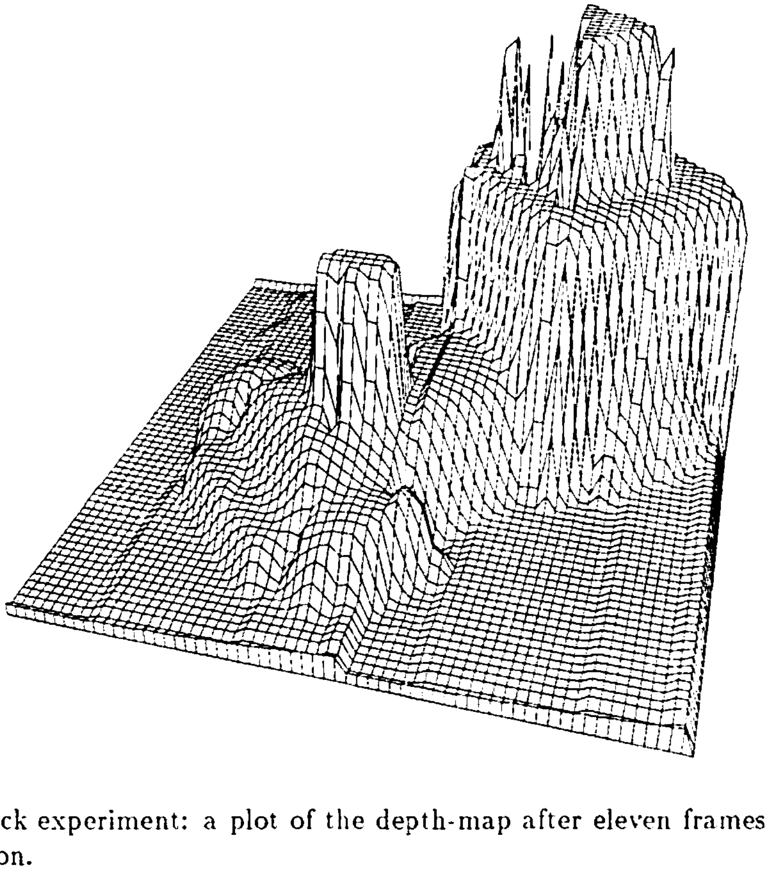


Figure 9: The toy-truck experiment: a plot of the depth-map after eleven frames using estimation-theoretic image-flow computation.

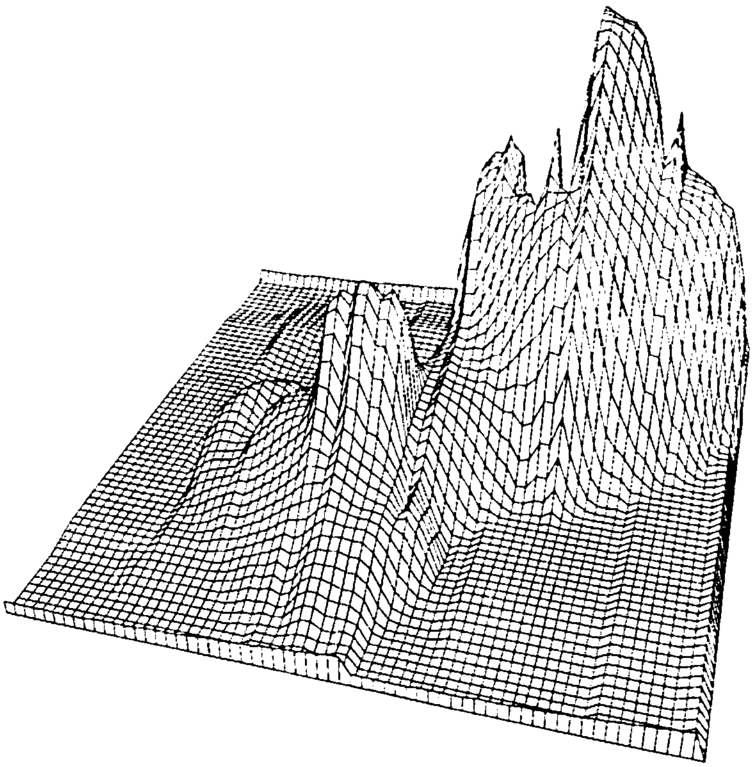


Figure 10: The toy-truck experiment: a plot of the depth-map after eleven frames using conventional smoothing-based image-flow computation.

Typically it is possible to know (or compute) the ground-truth flow-field only if (i) the motion is synthetically generated, e.g., by warping a given image in some known fashion or (ii) the camera motion and the depth of each point in the scene is exactly known. The second scenario is considered in the experiment that follows. The imagery for this experiement is selected in such a way that the flow-field does not have any discontinuities, simply because it is very difficult to come up with the ground-truth flow field in the presence of discontinuities.

Specifically, the scene is comprised of a textured poster rigidly mounted on a precision translation table. A 512×512 camera is mounted on the table as well, but its (translational) motion can be accurately controlled. The poster is placed facing the camera and slanted in such a way that (i) the optical axis is not perpendicular to the plane of the poster and (ii) the distance between the camera an the poster is very small (about 12 inches). Both these arrangements help to make the resulting flow-field interesting even when the camera is undergoing a pure translation. The camera is made to translate in a plane perpendicular to its optical axis so that the image displacement is roughly 6 pixels where the poster is closest to the camera and roughly 3 pixels where the poster is the farthest from the camera. The exact amount of camera translation as well as the distance of the lens from the rigid mount is recorded. The camera is then calibrated and its focal length

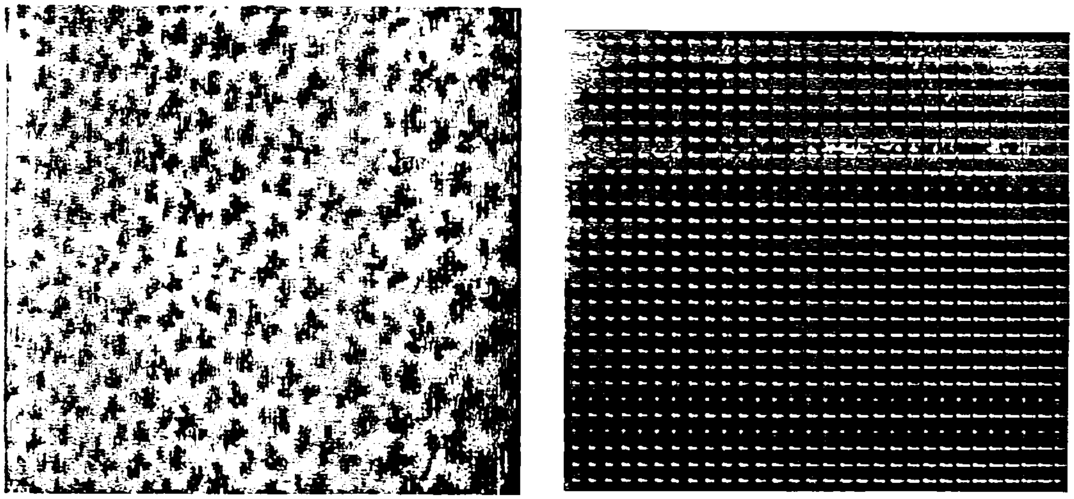

(a) (b)

Figure 11: The poster experiment: (a) central frame of the image-sequence and (b) correct flow-field.

is determined. The "correct" flow-field is determined using the theory developed by Waxman and Wohn [21]. The images are low-pass filtered and sub-sampled to get a resolution of 128 x 128 using Burt's technique [6]. Both components of image-velocity at each point are divided by four to get the correct flow-field corresponding to the reduced image size[1]. The central image and the correct flow-field are shown in Figures 11a and 11b respectively.

Two experiments are conducted, with correlation-window size set to 5 x 5 and 3 x 3 respectively. In each case, the percentage of pixels that have both components of velocity (a) within 5% (of the true value) (b) within 10% and (c) within 25%, before and after propagation (15 iterations), is determined. The results are shown in table 1. As expected, larger size of the correlation window (5 x 5) gives more accurate results, although reasonable results are obtained with a 3 x 3 correlation window also - specially after velocity propagation.

Figures 12 through 14 show various flow-fields and confidence measures obtained with the 3x3 correlation window. Figure 12a shows one frame of the original sequence. Figures 12b and 12c show the two confidence measures associated with conservation information (i.e., the "initial" estimate of velocity) at each point in the visual-field. These confidence measures are the inverses of the small and the large eigenvalue, respectively, of the covariance matrix S_{cc}. It is apparent that the one of the confidence-measures is high both at edges

[1] Actually, the reduced-size imagery will correspond to image-flow that is not exactly equal to the original image-flow reduced in magnitude by a factor of four. This is because of the intensity changes that accompany low-pass filtering and sub-sampling. Due to lack of a quantitative characterization of these changes, I do not account for them.

WINDOW SIZES ▽	Percentage of pixels with vector error less than 5%		Percentage of pixels with vector error less than 10%		Percentage of pixels with vector error less than 25%	
	Without Prop.	With Prop.	Without Prop.	With Prop.	Without Prop.	With Prop.
5 X 5 Search 3 X 3 Correl.	53.0%	56.1%	66.4%	77.5%	71.3%	83.1%
5 X 5 Search 5 X 5 Correl.	56.2%	61.2%	68.6%	81.6%	73.2%	86.4%

Table 1: Error statistics for the poster experiment. The two rows correspond to two different sizes of the correlation window. For each row, the first and the second columns indicate the percentage of total pixels for which the error in both components of velocity is less than 5% of the correct value, before and after velocity propagation respectively. The third and the fourth columns give the corresponding percentage of pixels with error less than 10%. Finally, the fifth and the sixth columns give the corresponding percentage of pixels with error less than 25%.

and corners of the intensity image whereas the other one is high only at corners. Figure 12d shows the initial estimate of the flow-field (i.e., the velocity U_{cc}). Figure 13 shows the flow-field after iterative velocity propagation (10 iterations). It is apparent that the flow-field is qualitatively correct almost everywhere in the image, except at a few randomly placed points. The velocity-estimate at these few points is incorrect because of a very high confidence associated with a wrong initial estimate (U_{cc}). As discussed earlier, such a situation can arise in some textured regions. Figures 14a and 14b show the two confidence measures after propagation.

Once again, in order to view the image-flow estimates obtained above in the context of depth-estimation, the procedure shown in appendix A is used to recover depth-maps. Eleven frames (shot at regular intervals as the camera translate horizontally by 0.5 inch, starting from the initial configuration described before, in a plane perpendicular to its optical axis) are used. Figure 15 shows the correct depth-map. Figures 16, 17 and 18 show the depth-map recovered by the procedure after three, seven and eleven frames respectively. Qualitatively, it is apparent that the depth estimates improve with time. Quantitatively, the root mean-square error in depth (over the entire image) is 11.2%, 4.3% and 2.8% after three, seven and eleven frames respectively.

In each of the two categories, the experiments reported here have small inter-frame motion. In order to

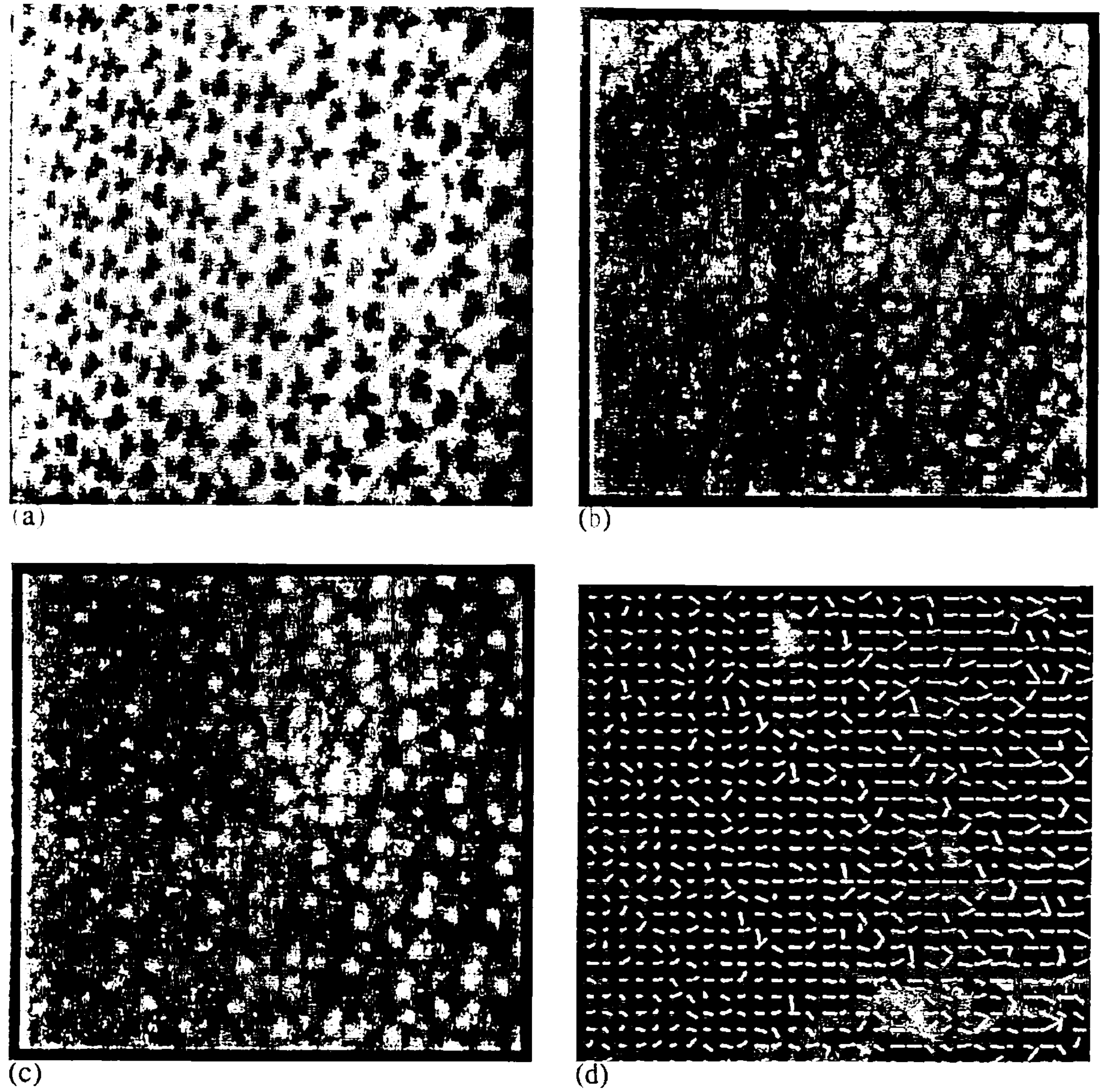


Figure 12: The poster experiment: (a) central frame of the image-sequence. (b).(c) confidence measures associated with conservation information. i.e.. the reciprocals of the eigenvalues of the covariance matrix S_{cc} and (d) initial estimate of velocity. i.e.. U_{cc}.

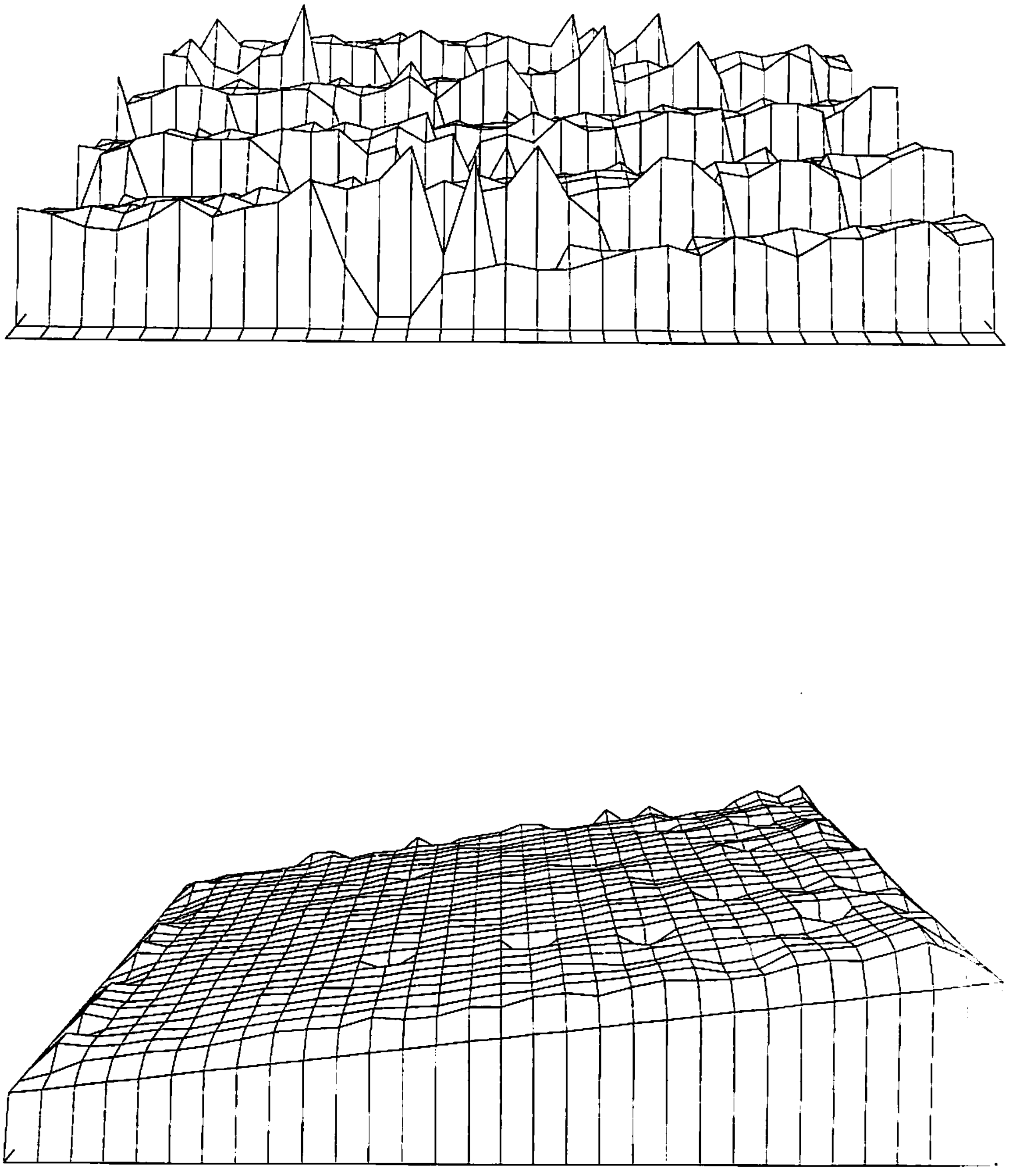


Figure 17: The poster experiment: depth-map after seven frames.

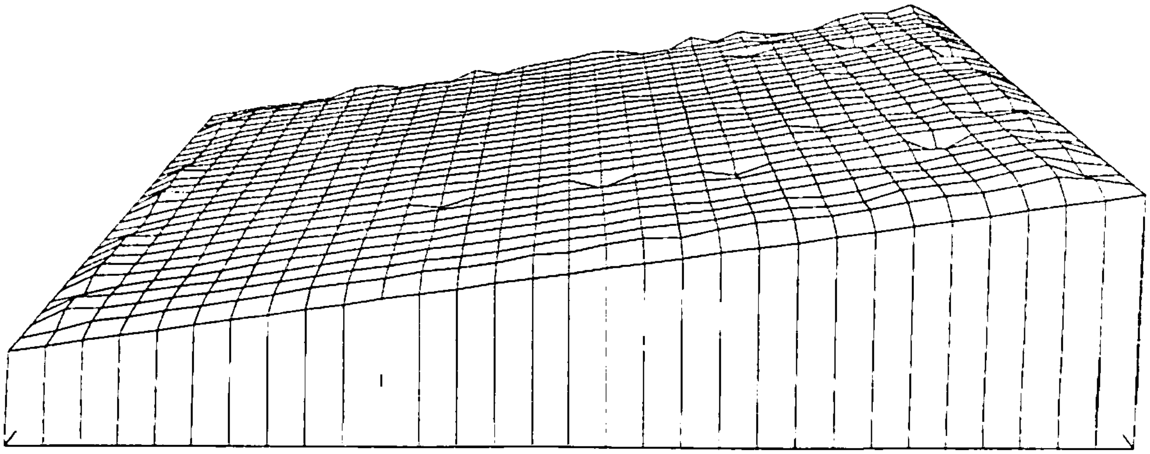


Figure 18: The poster experiment: depth-map after eleven frames.

handle the cases where motion can range from very small to very large, a hierarchical version of the algorithm has been developed based on the scheme proposed by Anandan [4]. The algorithm has been tested on a wide variety of scenes (including the famous dinosaur sequence used by Anandan, where velocity is of the order of eight pixels per frame) and it works very well. The results are not included here because of space limitations.

6 Conclusion

In this paper, I have shown a new framework for recovering image-flow from time-varying imagery. This framework recognizes the fact that velocity information available in small spatiotemporal neighborhoods in the imagery is not exact - there is uncertainty associated with it. It classifies the available information into two categories - conservation information and neighborhood information - and models each one of them using techniques that are common in estimation theory. It recovers the image-flow field by performing an optimal combination of the two types of information. Some of the distinctive features of the framework are summarized below.

1. It quantifies the velocity information contained in each of the two local sources - conservation and neighborhood - by an estimate and a covariance matrix. A similar approach has been used before by Anandan [4] for conservation information. However, as far as neighborhood information is concerned, this approach is novel. In essence, the current formulation accounts for the "spread" (in velocity space)

of neighborhood velocities in addition to their "average" that has been used in earlier formulations [10, 11].

2. It formulates the problem of estimating image-flow as that of performing a statistical combination of velocity estimates obtained from the two sources, on the basis of their covariance-matrices. The solution to this problem is iterative and amounts to propagating velocity information from regions of low uncertainty to regions of high uncertainty.

3. Because of the statistical nature of the procedure used to represent and propagate velocity, there is an explicit notion of confidence measures associated with the velocity estimate at each pixel, both before and after propagation. The idea of pre-propagation confidence measures has been used before [4] but that of post-propagation confidence measures is novel. The experiments shown in the previous section reveal that the iterative propagation procedure used in this framework does actually enhance the confidence during each iteration. The post-propagation confidence measure reflects the reliability of the final estimate of image-flow and it can be a valuable input to a system that uses image-flow to recover three-dimensional information. In the Kalman filtering-based depth estimation procedure used in this paper, the post-propagation variance (reciprocal of the confidence measure) serves as one of the inputs to the "prediction" stage.

4. The propagation procedure does a much better job of preserving the step-discontinuities in the flow-field, specially in the absence of texture in the vicinity of such discontinuities, as compared to the classic smoothing based propagation procedures [4, 11]. I have demonstrated this for the toy-truck sequence and the tori sequence in the previous section. Propagation procedures used in several frameworks proposed in the recent past [3, 10, 12, 15, 17, 21] are capable of preserving motion boundaries. However, the propagation procedure used in this framework is different from them in the following respects: (i) it gives image-flow in the entire visual-field, not just at the edges, (ii) it does not require any a-priori knowledge about the location of the boundaries, (iii) it does not assume that all intensity edges correspond to motion boundaries and vice versa, (iv) it does not use high order derivatives of the intensity function and (v) it is computationally simple.

There are several ways in which this framework can be extended and improved. Firstly, the behavior of response-distribution needs to be analyzed in greater detail, specially for the multimodal case. Secondly, in the current version of the framework, the velocity-propagation procedure utilizes only the estimate of velocity at neighboring pixels. It does not utilize the covariance-matrix associated with the estimate. It appears plausible that the knowledge of covariance-matrix might assist in identifying motion discontinuities, thus making the velocity-propagation procedure even more robust at discontinuities. Finally, the formulation of optimization problem assumes that conservation-error and neighborhood error are independent. In the current implementation, however, neighborhood information is derived from conservation information. This makes the two errors dependent. An investigation of the effects of this dependence will certainly be very useful in predicting the performance of the framework with respect to any given imagery. Also, efforts could be made to ensure that the two errors are, in fact, independent.

References

[1] E.H. Adelson and J.R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America.*, 2:284–299, 1985.

[2] J.K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images - a review. Technical Report TR-88-2-47, Computer Vision Research Center, University of Texas at Austin, 1988.

[3] J. Aisbett. Optical-flow with an intensity weighted smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5:512–522, 1989.

[4] P. Anandan. *Measuring Visual Motion from Image Sequences.* PhD thesis, COINS Department, University of Massachusetts, Amherst, 1987.

[5] J.V. Beck and K.J. Arnold. *Parameter estimation in engineering and science.* John Wiley and Sons, 1977.

[6] P.J. Burt. The pyramid as a structure for efficient computation. In A. Rosenfeld, editor, *Multi Resolution Image Processing and Analysis*, pages 6–37. Springer Verlag, 1984.

[7] B.F. Buxton and H. Buxton. Computation of optic flow from the motion of edge features in image sequences. *Image and Vision Computing*, 2, 1984.

[8] W. Enkelmann. Investigations of multigrid algorithms for estimation of optical flow fields in image sequences. *Computer Vision Graphics and Image Processing*, 43:150–177, 1988.

[9] D. Heeger. A model for extraction of image flow. In *First International Conference on Computer Vision*, 1987.

[10] E.C. Hildreth. *The Measurement of Visual Motion.* MIT Press, 1983.

[11] B.K.P Horn and B. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[12] J. Hutchinson, K. Koch, and C. Mead. Computing motion using analog and binary resistive networks. *Computer*, pages 52–63, 1988.

[13] L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image-sequences. In *Proceedings of the 2nd International Conference on Computer Vision, Tampa, FL*, pages 199–213, 1988.

[14] D.W. Murray and B.F. Buxton. Reconstructing the optic flow field from edge motion: an examination of two different approaches. In *First Conference on AI Applications, Denver*, 1984.

[15] H.H. Nagel. On the estimation of dense displacement maps from image sequences. In *Proceedings of ACM Motion Workshop, Toronto*, pages 59–65, 1983.

[16] A. Ralston and P. Rabinowitz. *A first course in numerical analysis*. McGraw-Hill Book Company, 1978.

[17] B. Schunck. Image flow: Fundamentals and algorithms. In J.K. Martin, W.N. Aggarwal, editor, *Motion Understanding: Robot and Human Vision*, pages 23–68. Kluwer Academic Publishers, 1988.

[18] G.L. Scott. *Local and Global Interpretation of Moving Images*. Morgan Kauffman Publishers, 1988.

[19] A. Singh. Image-flow estimation: An analytical review. Technical Report TN-89-085, Philips Laboratories, Briarcliff Manor, New York, 1989.

[20] M.A. Snyder. On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction. In *Proceedings of the IEEE Workshop on Visual Motion, 1989*, pages 107–115, 1989.

[21] A.M. Waxman and K. Wohn. Contour evolution, neighborhood deformation and global image flow: Planar surfaces in motion. *International Journal of Robotics*, 4:95–108, 1985.

[22] A.M. Waxman, J. Wu, and F. Bergholm. Convected activation profiles and measurement of visual motion. In *Proceedings of the IEEE CVPR, Ann Arbor, Michigan*, pages 717–722, 1988.

A Kalman filtering-based depth estimation from image-flow

Matthies, Szeliski and Kanade [13] had reported a Kalman filtering-based algorithm to recover dense depth maps from image-flow in the case of a stationary scene and known one-dimensional camera motion. This algorithm requires that an estimate of image-flow be produced along with its covariance for each new frame acquired (in a time-sequence) and be used to update the existing estimate of disparity (reciprocal of depth) and its variance. The principal advantage of such a scheme is that the uncertainty in depth estimates decreases with time. Matthies, et. al. had used Anandan's [4] smoothing-based algorithm to estimate image-flow and had performed error-analysis on the SSD surface to compute its variance. I have adapted their algorithm to use the framework for image-flow estimation discussed in this paper instead of Anandan's. Since this framework has an explicit covariance-matrix at each stage of computation, it fits into the Kalman filtering-based mechanism very naturally. Secondly, because of the discontinuity-preserving nature of the new framework, the discontinuities in the depth-field are better defined. This makes three-dimensional feature extraction (for interpretation of depth-fields) more reliable. Since the only modification to the original scheme of Matthies, et. al. is the way in which image-flow and its variance is estimated, the reader is referred to their original paper [13] for details of the procedure and its implementation. A block diagram of the modified

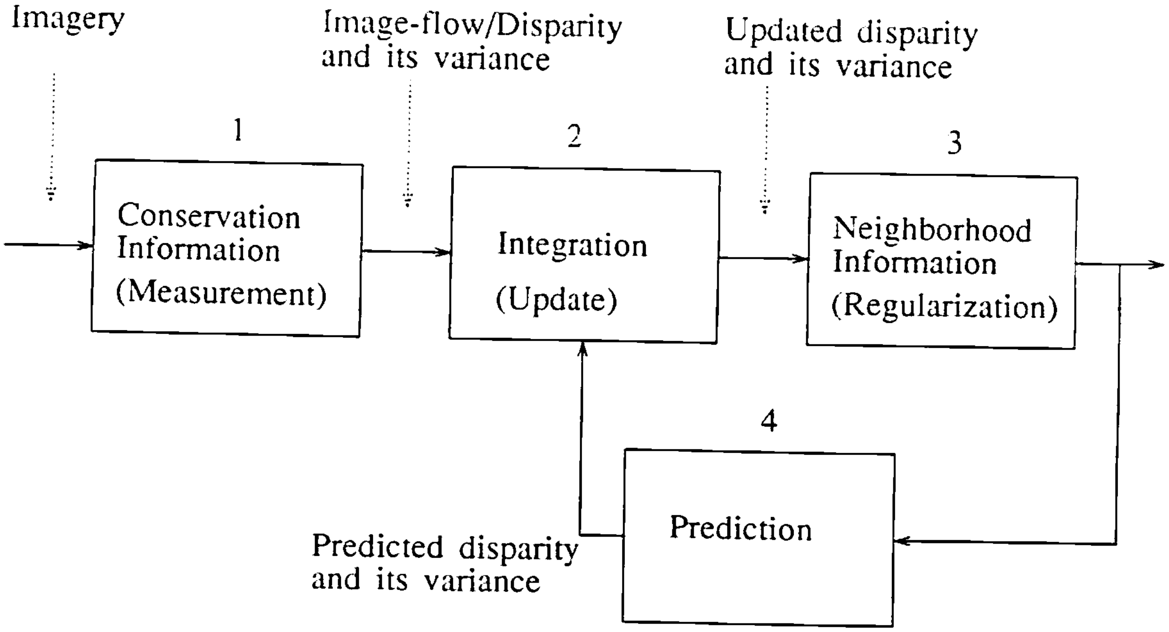

Imagery
Image-flow/Disparity and its variance
Updated disparity and its variance
1
2
3
Conservation Information (Measurement)
Integration (Update)
Neighborhood Information (Regularization)
4
Prediction
Predicted disparity and its variance

Figure 19: A block-diagram of the Kalman filtering-based depth estimation scheme.

scheme is shown in figure 19. The blocks 1 and 3 in this diagram depict the two steps of image-flow estimation and have been discussed in detail in this paper. The blocks 2 and 4 depict the "updating" and "prediction" steps of Kalman-filtering and are exactly the same as in [13].