MONOCULAR OPTICAL FLOW FOR REAL-TIME VISION SYSTEMS

STEPHEN M. BENOIT AND FRANK P. FERRIE

CENTER FOR INTELLIGENT MACHINES
MCGILL UNIVERSITY
MONTRÉAL, QUÉBEC
CANADA

e-mail: benoits@cim.mcgill.ca and ferrie@cim.mcgill.ca

ABSTRACT. This paper introduces a monocular optical flow algorithm that has been shown to perform well at nearly real-time frame rates (1 FPS) on natural image sequences. The system is completely bottom-up, using pixel region-matching techniques. A conjugate gradient descent method is broken down into two stages; pixel region matching error measures are locally minimized, and flow field consistency constraints apply non-linear adaptive diffusion, causing confident measurements to influence their less confident neighbors. Convergence is usually accomplished with one iteration for an image frame pair. Temporal integration and Kalman filtering predicts upcoming flow fields and figure/ground separation. The algorithm is designed for flexibility: large displacements are tracked as easily as sub-pixel displacements, and higher-level information can feed flow field predictions into the measurement process.

Subjects. Motion Analysis, Active and Real-Time Vision.

Key words and phrases. optical flow, region matching, flow consistency, real-time vision.

MONOCULAR OPTICAL FLOW FOR REAL-TIME VISION SYSTEMS

ABSTRACT. This paper introduces a monocular optical flow algorithm that has been shown to perform well at nearly real-time frame rates (1 FPS) on natural image sequences. The system is completely bottom-up, using pixel region-matching techniques. A conjugate gradient descent method is broken down into two stages; pixel region matching error measures are locally minimized, and flow field consistency constraints apply non-linear adaptive diffusion, causing confident measurements to influence their less confident neighbors. Convergence is usually accomplished with one iteration for an image frame pair. Temporal integration and Kalman filtering predicts upcoming flow fields and figure/ground separation. The algorithm is designed for flexibility: large displacements are tracked as easily as sub-pixel displacements, and higher-level information can feed flow field predictions into the measurement process.

Subjects. Motion Analysis, Active and Real-Time Vision.

Key words and phrases. optical flow, region matching, flow consistency, real-time vision.

MONOCULAR OPTICAL FLOW FOR REAL-TIME VISION SYSTEMS

1. Introduction

In this paper we describe a fast monocular optical flow algorithm for real-time applications that features rapid convergence (within a single iteration), ease of temporal integration, and swift reaction to abrupt change in scene motion. Although the flow data are represented on a coarse grid, the quantitative and qualitative flow for natural scenes is as good or better than algorithms in the same class.

The system is completely bottom-up, but does incorporate predictions from higher-level processes. In the current implementation a Kalman filter is used to predict upcoming flow fields and to perform figure/ground separation. The heart of the algorithm is a conjugate gradient descent method that alternately minimizes local correspondence errors and the consistency of adjacent flow field vectors. What results is a non-linear adaptive diffusion in which confident measurements are used to influence their less confident neighbours.

The principles used to formulate the algorithm follow directly from an approximate model of the hypercolumn organization present in the primate visual cortex. This model suggests a process in which scalar and vector information are processed independently and how they might be combined to produce a coarse flow field that is both accurate and robust. The optimization of region correspondences and flow field consistency, for example, are implemented as separate minimization stages as in the biological case instead of one lumped minimization.

The paper begins in Section 2 with a brief outline of the biological motivation for the algorithm and then briefly summarizes some of the related work. The algorithm itself is described in Section 3 along with details of our particular implementation. Experimental results presented in Section 4 demonstrate the performance of the algorithm on both synthetic and real data using some of the well-known data sets cited in the literature [5], and show that it it comparable to some of the best results obtained. The paper concludes in Section 5 with some final observations and pointers to future work.

2. Background

2.1. **Biological Clues.** The primate visual system is arguably the most developed vision system that can be studied. Computer vision research has much to gain from studying the structure and organization of the only fully-functional vision system architectures, developed not by engineers, but by nature.

Staining neurons of the visual cortex for cytochrome oxidase identifies clusters of neurons that are highly active, contrasted with clusters of lower activity [1]. The regions of higher activity (blobs) are found to be scalar representatives of the visual fields, while the lower activity regions (interblobs) are orientation detectors [1]. The scalar zones are more accustomed to continuous activity and react abruptly to changes in input, while the oriented zones fatigue rapidly unless the images shift, allowing the neurons to rest until activated again, responding smoothly to changes in input. This suggests that shifting the input images by eye movement becomes necessary not only to refresh the retinal cells, but also to refresh the oriented cells.

Some of the clues for region-based motion (and stereo vision) correspondence that are already widely-known include the alternating left- and right-occular dominance hypercolumns in the visual cortex. In each hypercolumn of the interblob (oriented) regions, different orientations and scales are represented, higher neural firing rates indicating a stronger presence of each element. In each hypercolumn of the blob (scalar, non-oriented) regions, different spacial scales of light intensity are represented.

The parvocellular pathway, dominated by orientation detectors, has been suspected of being largely responsible for matching edge information from left and right visual fields, and the medial temporal region for tracking this positional information over time. This would be a strong indication for the importance of edge information in sequential image analysis.

But what about the non-oriented, smoothly-varying or textured image fields that do not trigger the oriented-edge cells? Surely, scalar information would be more useful in these zones. The correspondence problem would be given another constraint to help the solution converge more rapidly, and reduce the solution space. The blob and interblob regions are interspersed, and can share information with each other. The blobs might perform image intensity region (or template) matching in the hypercolumns, and similarly, the interblobs might perform a comparison of flow orientations across several scales.

Because the blob regions respond abruptly to changes in input, neighboring neurons within the blobs may represent very different light intensities, as opposed to the interblob regions, where edge and flow orientations are strongly influenced by neighboring orientations. One would expect that within the blob regions, neighboring neurons do not diffuse their estimates as much as neurons within interblob regions. This characteristic could allow a form of window or region matching of light intensities or textures that takes place between corresponding receptive fields in the blob regions over time.

A key issue underlined by Yeshurun [9] in stereo and motion perception is the difference in size between a receptive field and the region represented by a hypercolumn. This observation implies that a motion patch in a hypercolumn represents changes from a cluster of neighboring receptive fields, and that the motion field will be less dense than the input image field.

The combination of these relevant facts suggest that the scalar and oriented hypercolumns perform different tasks to produce optical flow motion representation, and the results from both are combined to produce a coarse flow field. The blob-interblob interaction may be modelled as intensity-based region matching diffusing its estimates, and the orientation of the flow elements as a constraint or boundary for the diffusion.

2.2. **Other region-matching methods.** The comprehensive study by Barron et al. [4, 5] on the performance of optical flow techniques describes, classifies and compares representative algorithms using similar conditions. To borrow from their terminology, a region-based matching algorithm defines the velocity $\vec{v}$ as the shift $\mathbf{d}$ that yields the best fit between image regions at different times [5]. The methods of Anandan [2, 3] or Singh [7, 8] maximize a similarity measure, such as minimizing the sum-of-squared differences (SSD),

(1)

$$SSD_{1,2}(x, y; d_x, d_y) = \sum_{j=-n}^{n} \sum_{i=-n}^{n} W(i, j)\big[I_1(x + i, y + j) - I_2(x + d_x + i, y + d_y + j)\big]^2,$$

where W denotes a 2-D window function, and (d_x, d_y) are usually restricted to a small integer number of pixels. Alternatively, a distance measure minimization is added to the optimization, in the expectation that small displacements make more sense than larger displacements when the pixel patterns are matched equally well. Clearly, this is a bottom-up, or data-driven strategy that uses local information. The representative algorithms use different methods for reaching the optimal tile alignments, but neither use the rich information available at the flow geometry level, i.e. flow consistency constraints.

In this paper, we argue that the important information is not strictly a displacement that should be minimized, but rather, flow field consistency that should be enforced. By having the two layers of region matching and flow consistency constraints interact, we can gain significant advantages in stability, robustness and overall accuracy. For example in the case of textured surfaces with a repeating pattern the algorithm would be less likely to become trapped in a local minimum.

One might hope that applying a Gaussian diffusion operator to a flow field would somehow spread the flow field information throughout the moving object, but in reality, this only blurs the actual flow field. In order to be effective, flow field consistency must be enforced during the measurement stage.

3. Region-Based Optical Flow

As suggested in section 2, our algorithm performs region-based matching between successive image frames, at once minimizing a pixel pattern matching error and imposing flow field constraints within a neighborhood. For the purposes of this paper, we will consider the optical flow field to be a coarse field of image point correspondences between the two sequential images.

An overview of our algorithm is presented in the block diagram of Figure 1. Each stage is represented as one block, with the execution of stages proceeding from left to right. Each stage performs iterations internally, as indicated by the dark, curved arrows. Furthermore, there is a backward and forward exchange of flow data between the tile-matching and the flow consistency stages. Note that flow information from a Kalman prediction loop is used to predict the next set of region matching. This top-down feedback is used in our laboratory environment, but for this paper, we will be reporting results without using this high-level tracking and prediction, in order to compare our optical flow algorithm with other relevant algorithms. Each stage of this block diagram is described in more detail in the following sections.

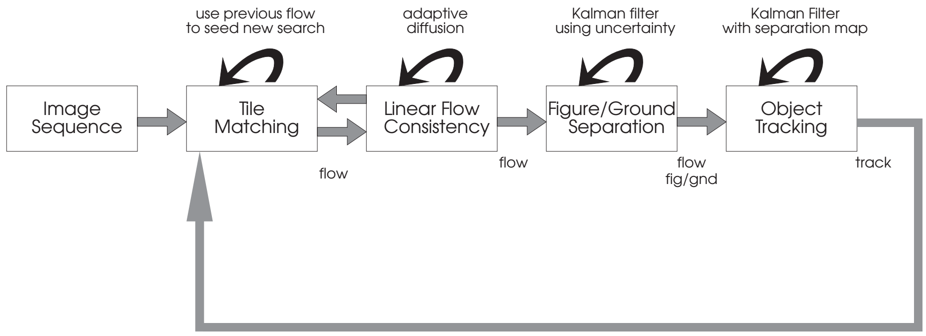


FIGURE 1. Block diagram of full optical flow process.

3.1. **Region Matching.** To find correspondences of clusters of pixels between the first and second images, our algorithm divides the first image into a grid of tiles, and proceeds to search for each tile's corresponding position in the second image, minimizing a pixel pattern-matching error metric between corresponding tiles.

The search for corresponding tile positions is assisted by providing an initial estimate of where each tile was predicted to move. This can be provided by a higher-level process in a larger vision system, and effectively tunes the measurement system to the expected motion events. For this paper, the predicted flow field is the flow field calculated from the preceding image pair.

For each tile p, there is a pixel pattern $P_{1p}(i,j)$ in frame 1 at position T_{1p} and a corresponding pattern $P_{2p}(i,j)$ in frame 2 at position T_{2p}. We define a difference and summation operation between the two corresponding tiles as

$$D_p(i,j) \triangleq \left\| P_{2p}(i,j) - P_{1p}(i,j) \right\| \tag{2}$$

$$S_p(i,j) \triangleq \left\| P_{2p}(i,j) + P_{1p}(i,j) \right\| \tag{3}$$

$$err_p(i,j) \triangleq \frac{D_p(i,j)}{S_p(i,j)} \tag{4}$$

$$err_p = \sum_{i,j} err_p(i,j) \tag{5}$$

This difference and summation are performed between corresponding pixels, and the resulting error term err_p for the tile summarizes the average pixel-matching error. The error function expresses a difference of intensities, normalized by their mean. To combat sensor noise, thresholds are applied to D and S, to clip unwanted behavior at sensor input extremes. This applies mainly to when the input intensities are very low, and governed by noise. When the summation of the pixel intensities is too small, the data are essentially unusable. Also, when the differences between successive inputs are very low, the intensities should be considered essentially the same.

$$\text{for } S_p(i,j) < S_\theta, \text{ set } err_p(i,j) = err_{max} \tag{6}$$

$$\text{for } D_p(i,j) < D_\theta, \text{ set } err_p(i,j) = err_{min} \tag{7}$$

The advantage of using these two constraints becomes clear when using natural scenes encoded by video cameras: even in a static scene, digitization or other noise can introduce speckles or streaks in an image sequence that most algorithms would prefer chasing. These parameters were chosen to be $S_\theta = 16$, $D_\theta = 8$, for the intensity thresholds of a 256-level digitized image. The error levels were never allowed to be exactly 1.0 or 0.0. Instead, we preferred the more numerically stable choices of $err_{max} = 0.99$ and $err_{min} = 0.01$.

3.2. **Flow Field Consistency.** Flow field consistency is the behaviour of a flow field that obeys the constraints suggested by psychophysical experiments in motion perception. For example, texture flow fields are improved when the measurement process includes a texture flow curvature consistency constraint [6]. The issue of how to measure or enforce optical flow field consistency now deserves attention. Applying curvature consistency would probably improve the optical flow field, but due to the coarse sampling of tile alignments and equally coarse directional encoding, a linear flow consistency is more appropriate.

Linear flow consistency implies that patches of an image should be moving in roughly the same direction as their neighbors. In cases of uncertainty, when a patch is moving in a direction contrary to all its neighbors, the neighbors will influence the outlier more than vice versa. An easily-implemented updating rule performs a weighted averaging of neighbor's displacements, each contribution weighted by a similarity measure. This similarity measure encodes a similarity of direction and magnitude between two given vectors.

Adaptive diffusion allows confident neighbors to influence uncertain tiles without affecting already confident tiles. To apply linear velocity consistency between all adjacent tiles, we define $\vec{v}_p$ as displacement of tile p between frames 1 and 2, i.e. $T_{2p} - T_{1p}$.

At each iteration k,

$$\vec{v}_p^{k+1} \leftarrow \frac{\sum_{n \in \mathbf{N}} w_n^k \vec{v}_n^k}{\sum_{n \in \mathbf{N}} w_n^k}, \tag{8}$$

where n is a neighbor of the tile from neighborhood N, and w is a weighting function that measures the similarity between a tile's motion vector and its neighbor n's motion vector.

The requirements of linear flow consistency call for function w to return a high weight when the vectors were similar, and a low weight when they were dissimilar. In this situation, both the magnitude and direction similarity are considered equally important. The similarity measure is divided into two components, magnitude similarity $S_m(\vec{v}_1, \vec{v}_2)$ and direction similarity $S_d(\vec{v}_1, \vec{v}_2)$.

$$S_d(\vec{v}_1, \vec{v}_2) = \begin{cases} \frac{1+|\vec{v}_1 \cdot \vec{v}_2|}{2|\vec{v}_1||\vec{v}_2|} & \text{when } |\vec{v}_1||\vec{v}_2| \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

$$S_m(\vec{v}_1, \vec{v}_2) = \begin{cases} 1 - \frac{|\vec{v}_1 - \vec{v}_2|}{|\vec{v}_1|+|\vec{v}_2|} & \text{when } |\vec{v}_1| + |\vec{v}_2| \neq 0, \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

Both functions have values ranging from 0.0 to 1.0. The overall similarity $S(\vec{v}_1, \vec{v}_2)$ is the linear combination

$$S(\vec{v}_1, \vec{v}_2) = \frac{1}{2} S_m(\vec{v}_1, \vec{v}_2) + \frac{1}{2} S_d(\vec{v}_1, \vec{v}_2) \tag{11}$$

This form of weighted averaging is a very non-linear diffusion process. The amount of diffusion between any two neighbors is governed by their local beliefs, in contrast to uniformly-weighted diffusion, which blurs away discontinuities in flow. After an update is performed for each tile, the tile pixel matching error is recomputed in one pass.

The usefulness of the linear flow consistency constraints become clear in cases of motion involving repeating textures. When region matching becomes ambiguous, the flow consistency constraints dominate. This is in opposition to region-matching schemes that embed a displacement-minimization constraint, which would tend to halt patches with ambiguous region matching.

3.3. **Figure/Ground Separation.** A moving object can be isolated from its background by associating a probability that a tile is observing image motion for each tile. Applying a Kalman filter to each tile's probability of occupation provides stability to this representation. For each frame pair k, the weight w_k is the number of tiles experiencing motion, i.e. those tiles moving faster than a threshold v_θ, typically 0.5

pixels / frame. Thus, the probability of occupation P_{occ} for tile p can be expressed as

$$P_{occ}^{k+1}(p) \leftarrow \frac{\hat{P_{occ}}^{k+1}(p)w^{k+1} + P_{occ}^{k}(p)w^{k}}{w^{k+1} + w^{k}}, \tag{12}$$

where $\hat{P_{occ}}$ is a measure of confidence that a tile is in motion. The end result is that a map of image regions is produced, isolating moving objects from the stationary background, integrated over time.

3.4. **Implementation Issues.** Note that the region matching and flow field consistency constraints could have been implemented as one error function to minimize. It can be shown that by alternatively measuring the region-matching error and enforcing the flow field consistency, the end effect is to perform a conjugate gradient descent locally for each tile, while diffusing measurements to neighboring tiles.

But by decoupling the stages as is evidenced by the primate visual cortex architecture, we achieve brief steps that can be implemented compactly and executed quickly. This way, each iteration is brief, and can either be repeated over the same image pair, or pipelined to another processing stage while new information is gathered. Fast implementation of the optical flow algorithm becomes possible.

The tiles were uniformly distributed over the image, and tests were performed using overlapping arrangements and non-overlapping arrangements, with various tile sizes, ranging from 3×3 pixels to 8×8 pixels, with 4×4 yielding a reasonable tradeoff of time to compute versus quality. During the region-matching stage, the algorithm tests a fixed number of tile displacements, searching around the predicted correspondence, but also testing for the case of sudden stopping. The latter case occurs most often when an object in the image sequence translates a distance the dimension of a tile. At one instant, the tile sees the object; at the next, the background.

4. Experimental Results

Five image sequences are presented here, consisting of two synthetic sets, two well-known natural image sequences, and one image sequence typical of the algorithm's intended environment.

4.1. **Translating Sinusoids.** This data set was obtained from the Barron et al. archive, and consists of the superposition of sinusoids. Error here is reported using the same error metric as reported in [4] and [5], namely the angular deviation from the correct flow direction. Representing the velocities as 3-d space-time unit direction vectors, $\vec{v} \equiv \frac{1}{\sqrt{v_1^2+v_2^2+1}}(v_1, v_2, 1)^T$, the error between the correct velocity $\vec{v}_c$ and an estimate $\vec{v}_e$ is

$$\psi_E = \arccos(\vec{v}_c \cdot \vec{v}_e) \tag{13}$$

The tests were performed on the *mysineB-6* (**Sinusoid 1**) and *mysineC-16* (**Sinusoid 2**) data sets, where the motions of the entire image plane are known to be (1.583, 0.863) pixels / frame and (1.0, 1.0) pixels / frame, respectively. The algorithm used a grid of 10×10 tiles, each tile of 6×6 pixels, applying 5 iterations between each image pair. The results shown are accumulated over the entire image sequence, not just a single frame pair. The results for our algorithm and those of Anandan and Singh are summarized in table 1. Sample frames and generated flow fields from the two sequences are shown in Figures 2 and 3.

Our algorithm thus responds very strongly to this class of stimulus, namely uniform translations. Note that the displacements for **Sinusoid 1** are not integer displacements, and rival other region-matching methods. Our algorithm's success for this class of input can be explained by the flow field consistency enforcement. The local information provided by region matching is propagated to neighbors who improve

Technique	Average Error	Standard Deviation
Us	5.21°	$< 10^{-10\circ}$
Anandan	30.80°	5.45°
Singh ($n = 2,\ w = 2,\ N = 2$)	2.24°	0.02°
Singh ($n = 2,\ w = 2,\ N = 4$)	91.71°	0.04°

TABLE 1. Results of **Sinusoid1** test data. Experimental results for Anandan and Singh are taken from [4] and [5].

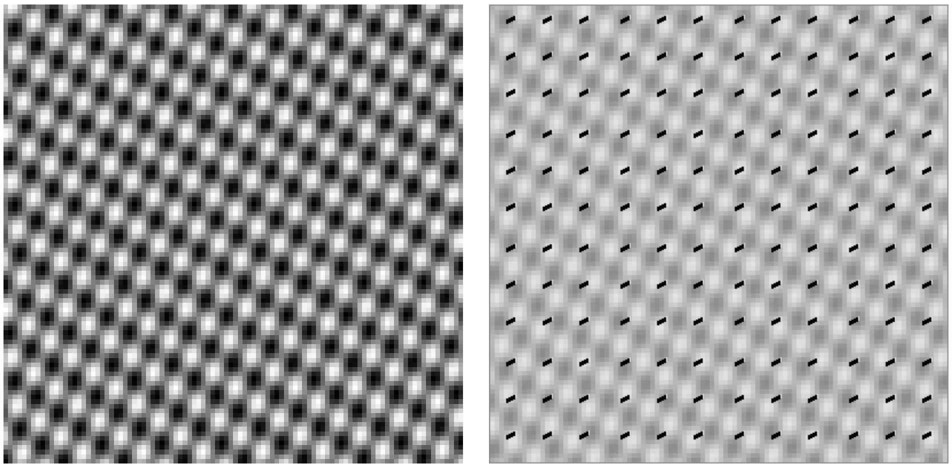
Technique	Average Error	Standard Deviation
Us	0.0452°	0.3607°
Anandan	-	-
Singh	-	-

TABLE 2. Results of **Sinusoid2** test data. Experimental results for Anandan and Singh are unavailable from [4] and [5], but are described as "unchanged".

their estimates with the new information. With weighted averaging of neighbors, non-integer displacements can be obtained despite the integer-based region-matching.

4.2. **Yosemite Fly-Through Sequence.** The Yosemite sequence, created by Lynn Quam, was chosen as a complex text case with a range of velocities, occluding edges and severe aliasing [4]. A frame of the sequence and the correct flow field appear in Figure 4. This experiment used a grid size of 80×60 tiles, each tile consisting of 8×8 pixels. Five iterations were performed on each frame pair.

The sequence was tested in two ways, first using every flow vector, regardless of confidence, and the second time, vectors falling above an error threshold were ignored. This is made possible by the measurement of region-matching error during



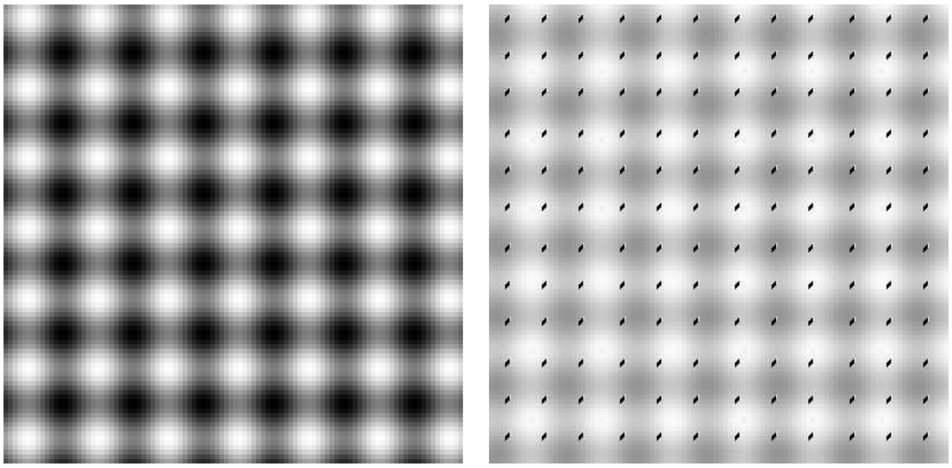
(a) (b)

FIGURE 2. **Sinusoid 1.** An image frame from the sequence **(a)**, and superimposed reconstructed flow field, **(b)**. Note that the flow image has been subdued for pictorial purposes only.

the minimization. The applied error threshold was 0.025, and affected 32.4% of the flow vectors. The results for the thresholded and unthresholded experiments are summarized in table 3.

More details on the experiment are shown in Figures 5 and 6. These show the measured flow fields and error data. Note that our algorithm is competitive with the algorithms of Anandan and Singh. As explained earlier, our algorithm represents flow information as a coarse data set, versus the conventional dense data set (represented in table 3 as percentages of the image surface used. In particular, our algorithm has a tighter distribution of low-error flow data than either Anandan or Singh in most cases. We concede, of course, that Singh's mean error and standard deviation is better than ours in the thresholded case.

What is not shown here is the qualitative performance of Anandan's or Singh's algorithms on this image sequence. There are outliers in both algorithms' flow fields

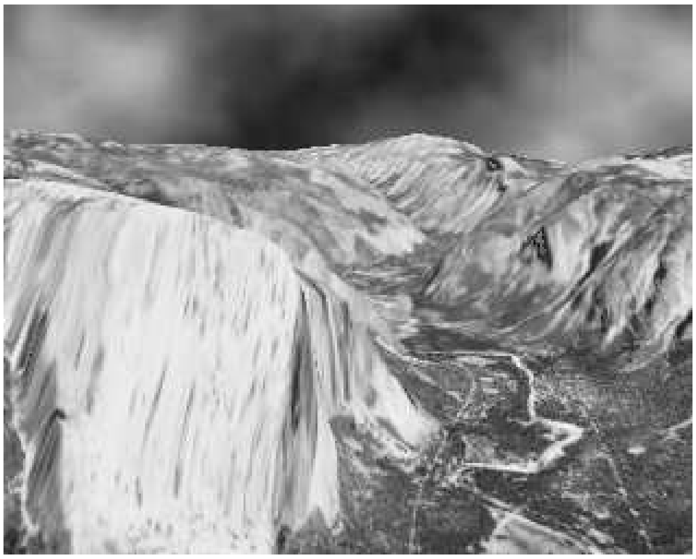


(a) (b)

FIGURE 3. **Sinusoid 2.** An image frame from the sequence **(a)**, and superimposed reconstructed flow field, **(b)**. As before, the flow image has been subdued for pictorial purposes only.

Technique	Valid Data	Average Error	Std. Dev.	$< 1°$ Error	$< 2°$ Error	$< 3°$ Error
Us Unthresholded	4800	17.16°	17.50°	1.96%	7.25%	13.35%
Us Threshold=0.025	3245	15.13°	15.57°	2.43%	9.37%	16.86%
Anandan	100%	13.46°	15.64°	1.1%	4.1%	8.0%
Singh (st 1, $n = 2$, $w = 2$)	100%	15.28°	19.61°	1.3%	3.7%	7.0%
Singh (st 1, $n = 2$, $w = 2$, $\lambda_1 \leq 6.5$)	11.3%	12.01°	21.43°	12.3%	24.4%	34.6%
Singh (st 2, $n = 2$, $w = 2$)	100%	10.44°	13.94°	-	-	-
Singh (st 2, $n = 2$, $w = 2$, $\lambda_1 \leq 0.1$)	97.7%	10.03°	13.13°	2.4%	7.4%	12.6%

TABLE 3. Results of Yosemite test data. Mean and standard deviation experimental results for Anandan and Singh are taken from [5], while the low angular error distribution were obtained from [4].



(a)

(b)

FIGURE 4. **Yosemite Sequence.** An image frame from the sequence **(a)**, and correct flow field, **(b)**.

that are not allowed to occur in our algorithm. This qualitative information can be found in [5].

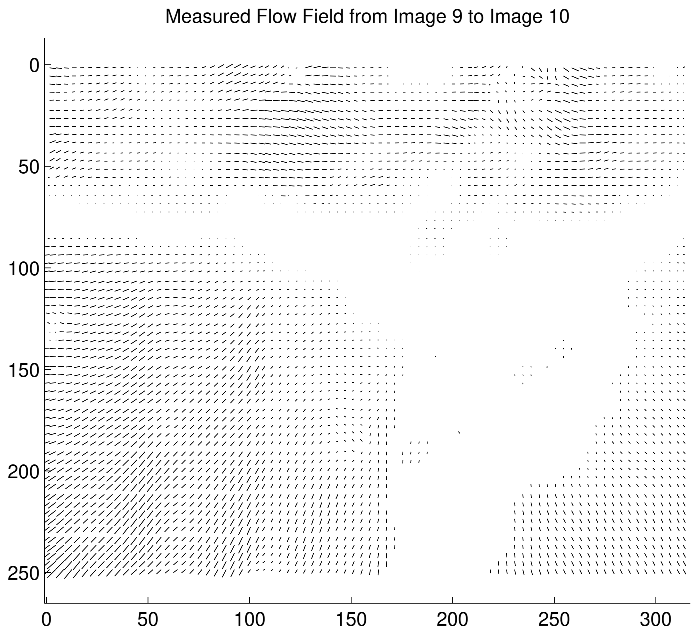


FIGURE 5. **Yosemite Sequence.** Measured flow field

4.3. **Hamburg Taxi Sequence.** The Hamburg taxi sequence has four principal moving objects, including a taxi turning the corner, a car in the lower left moving from left to right, and a van in the lower right moving from right to left. A pedestrian is also walking on the sidewalk in the upper left, but the motion was too far below the error threshold for our algorithm to detect. Alongside the image of the flow field superimposed on the scene in Figure 7 are velocity and occupation maps. The algorithm used only one iteration per image pair, with a grid of 80×60 tiles, each tile at 6×6 pixels. For clarity, an image frame and the flow field are shown separated in Figure 8.

Qualitatively, the background is shown to be immobile, despite the large amount of white noise and aliasing present in the image sequence. Anandan's output does not show the vertical displacement of the taxi, while Singh's output shows less coherent motion (without thresholding) and noise along the bottom of the image frame where there is no motion.

4.4. **SRI Tree Sequence.** This is a low-contrast image sequence, where the camera translates perpendicularly to the line of sight. There is a large amount of occlusion, and the highest image velocities were found to be just under 3 pixels per second. A sample frame of the image sequence and the measured flow field are shown in Figure 9. In this special case of camera translation, we can use the kinetic depth effect (proximity proportional to velocity) to show an approximate depth map of the scene, shown in Figure 10.

The algorithms of both Anandan and Singh do reasonably good jobs on the SRI tree sequence. But both have discontinuous flow fields in locations where the motion is fluid, whereas our output has a consistent flow field.

4.5. **Hand-held target.** This scene is typical of the events we wish to measure. An end user presents a target object to the workstation's video camera and moves the object while viewing the result on-screen in real-time. The rich flow field (see Figure 11) will be used in later stages for qualitative shape description. This scene demonstrates the algorithm under its best conditions, i.e. low camera noise and high-contrast textures. As in all the above examples, only one iteration of the algorithm was applied to each image pair. Image velocities for this particular frame pair approached 5 pixels / frame, but velocities as high as 10 pixels / frame have been successfully tracked. As before, a relief map is shown in Figure 12 to illustrate the crisp boundaries of the target object and coherency of the flow field magnitude.

5. Conclusions

The results obtained using our algorithm suggest that the biologically-motivated strategy of interleaving scalar region correspondence with flow field consistency operations leads to a stable inference of optical flow that can serve as a stable basis for further interpretation. Performance is comparable to the best algorithms in terms of both quantitative and qualitative performance with the additional advantages of speed and adaptability. The algorithm is also flexible - large displacements are tracked as easily as sub-pixel displacements, and high-level information can feed flow field predictions into the measurement process (e.g. Kalman filtering).

One of our interests is in the application of optical flow analysis to 3-D shape recovery. The example shown in Figures 11 and 12 shows how the algorithm can be used to recover the shape of an object waved in front of the camera. Motion segmentation clearly delineates the figure/ground separation and the integration of the resulting flow field leads to the recovery of the visible surfaces. Further improvements are possible by fitting a parametric model to the accumulated data and predicting upcoming flow fields by projecting the object model's motion onto the image plane, accelerating the acquisition process and improving the quality of the flow data. With continuing improvements in flow estimation and computing technologies, practical machine vision applications based in optical flow analysis should find increasing popularity.

References

[1] J. Allman and S. Zucker. Cytochrome oxidase and functional coding in primate striate cortex: An hypothesis. *Cold Spring Harbor Symp. Quant. Biology*, 55:979–982, 1990.

[2] P. Anandan. *Measuring Visual Motion from Image Sequences.* PhD thesis, Univ. of Massachusetts, Amherst, MA, 1987. COINS TR 87-21.

[3] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, (2):283–310, 1989.

[4] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. Technical report, Robotics and Perception Laboratory, Department of Computing and Information Science, Queen's University, Kingston, Ontario, July 1992.

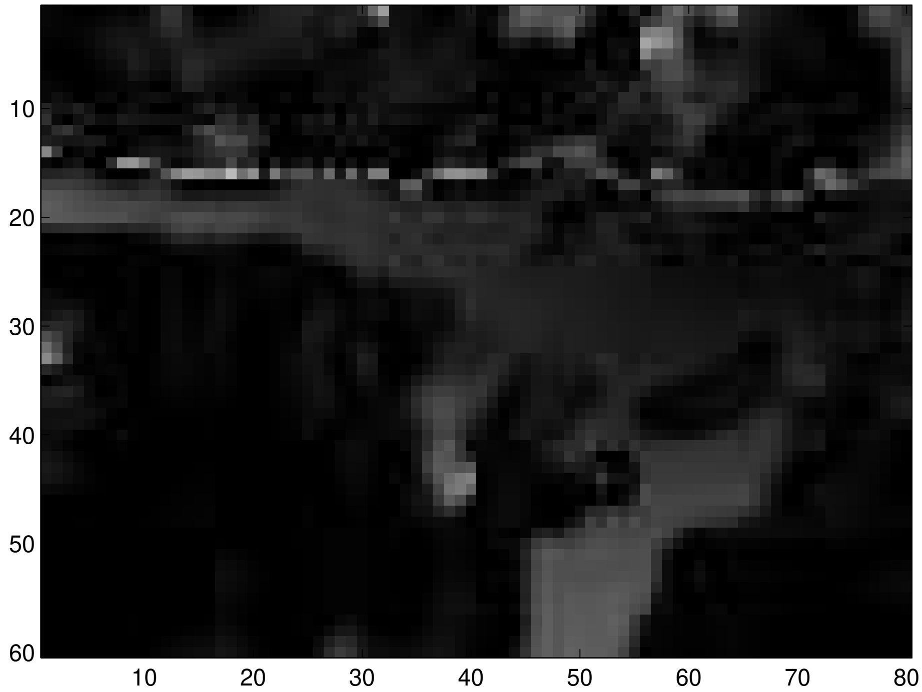
[5] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.

[6] P. Parent and S. Zucker. Curvature consistency and curve detection. *J. Opt. Soc. Amer., Ser. A*, 2(13), 1985.

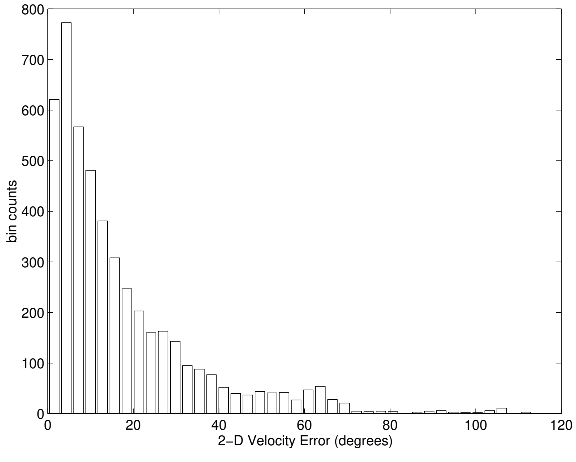
[7] A. Singh. An estimation-theoretic framework for image-flow computation. Proc. IEEE ICCV, 1990. Osaka, pp. 168-177.

[8] A. Singh. *Optic flow computation: a unified perspective.* IEEE Computer Society Press, 1992.

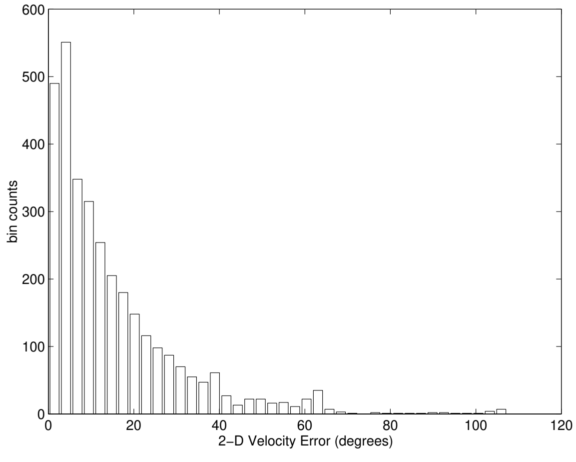
[9] H. Yeshurun. Size limits on stereo and motion perception: Back to the hypercolumn? Lecture, October 1995.



(a)



(b)

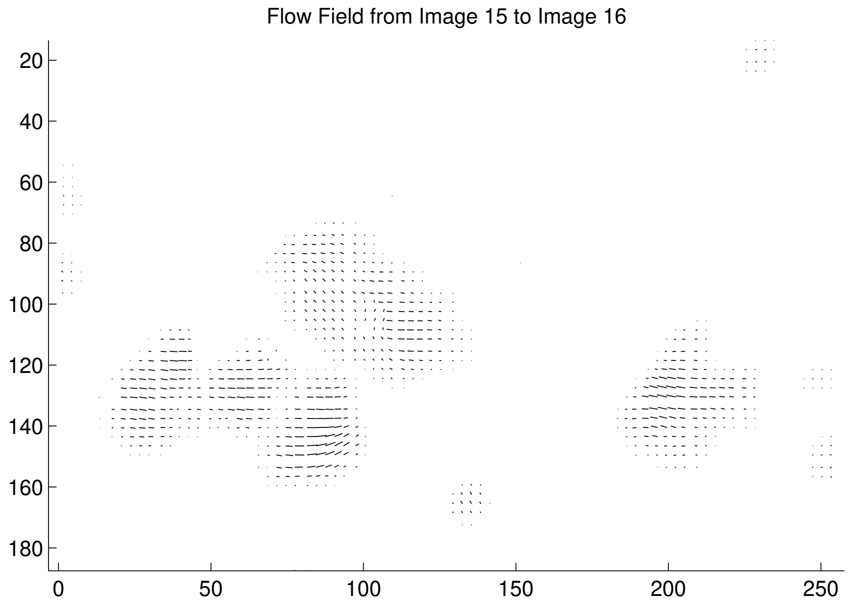


(c)

FIGURE 6. **Yosemite Sequence.** An angular error image for frame 10 **(a)**: black indicates zero error, white is 180° error. Also shown are angular error distribution for the unthresholded **(b)** and thresholded **(c)** experiments.

FIGURE 7. **Hamburg Taxi Sequence, workstation view.** The flow magnitude is rendered in the upper right window as intensity proportional to image velocity. In the lower right window, the figure/ground separation is rendered as a binarized image.

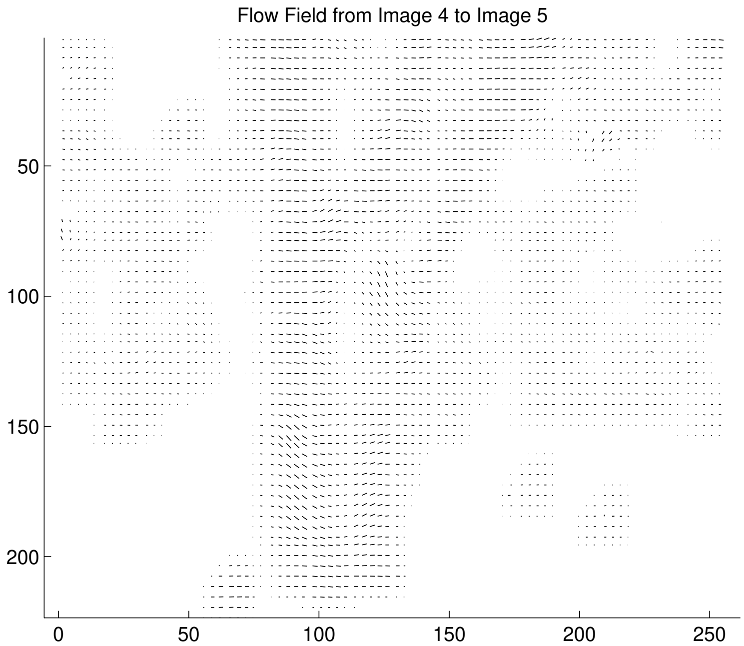
(a)



(b)

FIGURE 8. **Hamburg Taxi Sequence.** Frame 15 from the image sequence is shown in **(a)**. The flow field between frames 15 and 16 is shown in **(b)**.

(a)



(b)

FIGURE 9. **SRI Tree Sequence.** Frame 5 from the image sequence is shown in **(a)**. The flow field between frames 4 and 5 is shown in **(b)**.

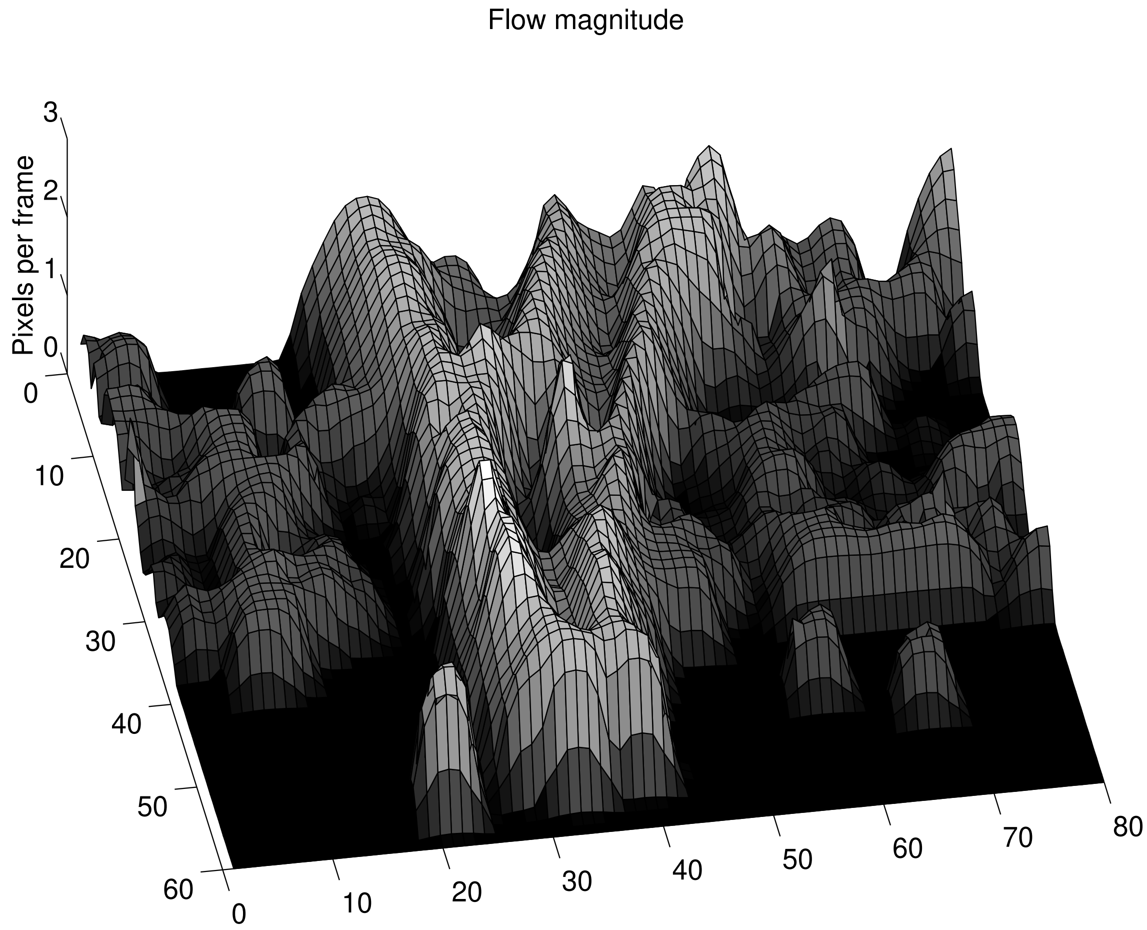
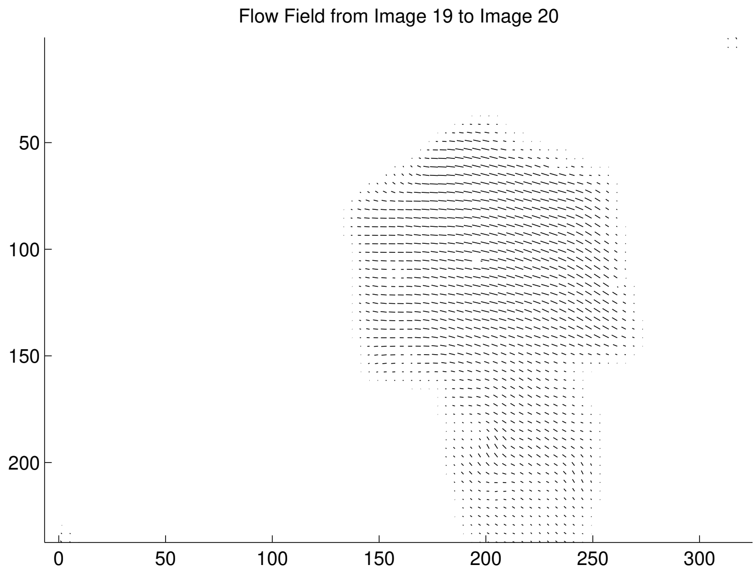


FIGURE 10. **SRI Tree Sequence, Kinetic Depth.** The magnitude of the flow field is rendered here as a relief map.

(a)



(b)

Figure 11. **Hand-Held Target Sequence.** Frame 20 from the image sequence is shown in **(a)**. The flow field between frames 19 and 20 is shown in **(b)**.

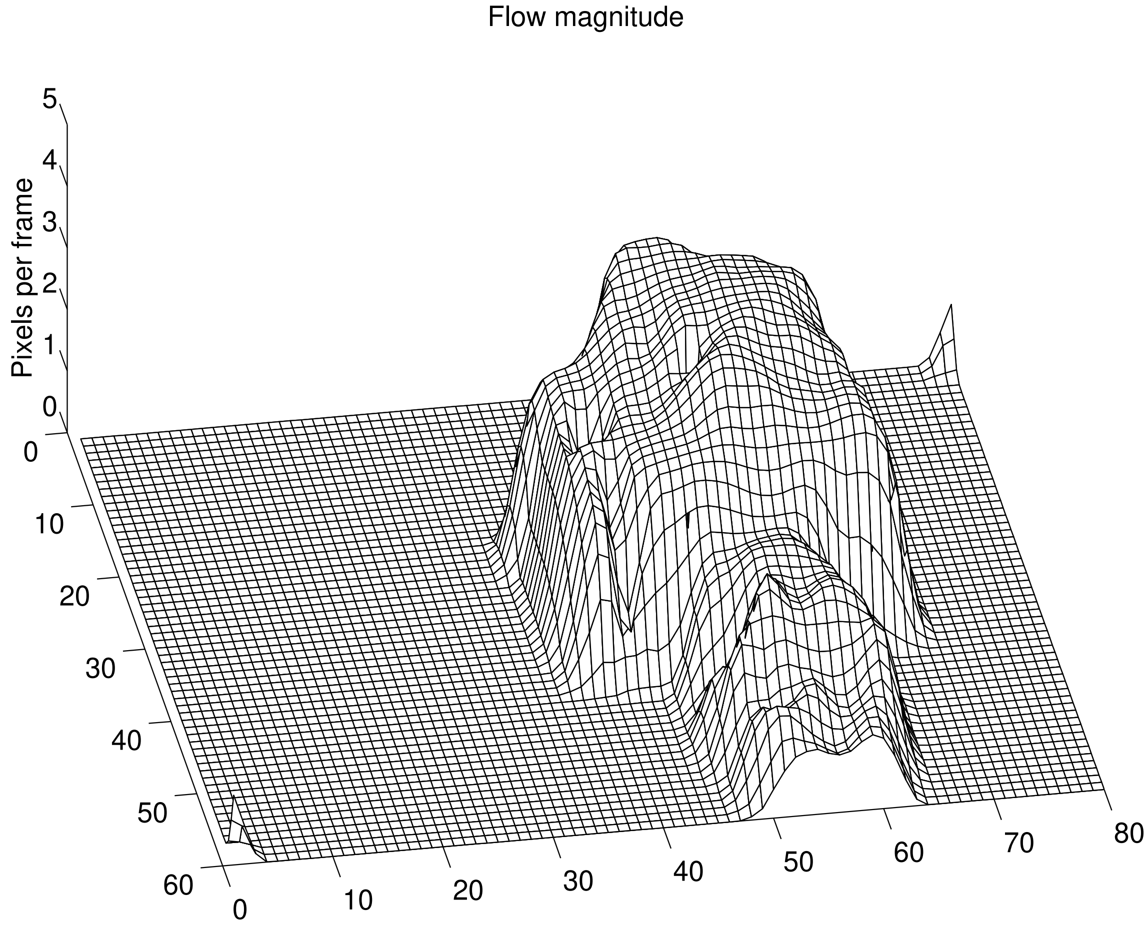


Figure 12. **Hand-Held Target Sequence, Kinetic Depth.** The magnitude of the flow field is rendered here as a relief map.