Computer Vision and Image Understanding **84,** 126–143 (2001)
doi:10.1006/cviu.2001.0930, available online at http://www.idealibrary.com on IDEAL®

On Benchmarking Optical Flow

B. McCane

Department of Computer Science, University of Otago, Dunedin, New Zealand
E-mail: mccane@cs.otago.ac.nz

K. Novins

Department of Computer Science, University of Auckland, Auckland, New Zealand
E-mail: novins@cs.auckland.ac.nz

and

D. Crannitch and B. Galvin

Department of Computer Science, University of Otago, Dunedin, New Zealand

Received September 12, 1999; accepted August 8, 2001

Evaluating the performance of optical flow algorithms has been difficult because of the lack of ground truth data sets for complex scenes. We present a new method for generating motion fields from real sequences containing polyhedral objects and present a test suite for benchmarking optical flow algorithms consisting of complex synthetic sequences and real scenes with ground truth. We provide a preliminary quantitative evaluation of seven optical flow algorithms using these synthetic and real sequences. Ultimately, we feel that researchers should benchmark their own algorithms using a standard suite. To that end, we offer our Web site as a repository for standard sequences and results. © 2001 Elsevier Science (USA)

Key Words: optical flow; quantitative evaluation.

1. INTRODUCTION

Despite the volumes of research on the topic of optical flow, few attempts have been made to empirically evaluate the performance of optical flow algorithms on complex image sequences. A major obstacle in performing any empirical study in computer vision is obtaining ground truth data. Until quite recently, the majority of evaluations have been purely qualitative or utilize very simple synthetic scenes from which ground truth is easy to extract. Although such evaluations can be useful in a relative sense, they do not give a good indication of the absolute quality of any particular algorithm.

The only major previous work in the area of quantitative optical flow evaluation is the work of Barron *et al.* [5] and Otte and Nagel [16]. Barron *et al.* conducted an empirical and qualitative analysis of nine optical flow algorithms. The algorithms were tested on five synthetic image sequences where ground truth motion fields were available and four real image sequences for which no ground truth motion field was available. In the four real image sequences, no objective evaluation was performed. Four of the synthetic sequences were constructed from simple 2D image operations, for example, translating a simple pattern. The final synthetic scene was more complex. It shows a fly-through of the Yosemite National Park; however, there is little occlusion, and none of the complex lighting effects that occur in real scenes. While this important work has had a great impact on the field, it does suffer from some deficiencies. First, the synthetic image sequences satisfied many of the unrealistic assumptions of most optical flow algorithms. Second, the real image sequences were only evaluated in a qualitative manner.

Otte and Nagel [16] present a good evaluation of their optical flow algorithm by comparing it against others using one of the synthetic sequences from Barron *et al.* They also perform a qualitative comparison using one of their own sequences. Their paper is significant because they actually measured the ground truth motion field for their real sequence and have made the sequence and the motion field publicly available. While their method for measuring the motion field is general and thorough, it is quite complex and requires significant hardware capabilities. The method requires an accurate 3D model of the scene, a calibrated camera, and knowledge of the trajectory of the camera and objects throughout the sequence. In their work the camera is mounted on a robot arm undergoing known motion, and object motion is presumably physically measured from the scene. These limitations have restricted the use of the method by other researchers.

In this paper, we hope to redress the deficiencies of both Barron *et al.* and Otte and Nagel's empirical work. The goal of this paper is to present a benchmarking suite of image sequences and tools for the purpose of evaluating optical flow algorithms. To that end, we provide a comprehensive set of complex synthetic scenes, a simple interactive method for extracting ground truth motion from real polyhedral scenes, and three new ground truth motion data sets from such scenes. We have found that our synthetic scenes are useful for determining how well an algorithm scales to increasing complexity and for measuring the relative performance of different algorithms. However, for a true measure of performance, evaluation on real scenes is needed and the three example real scenes are provided to demonstrate the technique.

We do not claim to be providing an evaluation of the current state of the art in the field, although we do revisit the evaluations performed by Barron *et al.* [5]. We hope that researchers who develop new optical flow algorithms will make use of our data to compare their algorithms with already established ones. To facilitate this, we have constructed an online Web page that provides free access to our raw data, tools, and evaluations [1].

2. WHAT IS GROUND-TRUTH OPTICAL FLOW?

Horn and Schunck [8] define optical flow as follows:

The optical flow is a velocity field in the image that transforms one image into the next image in a sequence. As such it is not uniquely determined; the motion field, on the other hand, is a purely geometric concept, without any ambiguity—it is the projection into the image of three-dimensional motion vectors.

In practice, specular effects, shadows, insufficient texturing, and occlusion make the unambiguous recovery of the true motion field impossible. If these effects are present to sufficient degree, the optical flow vectors produced are likely to be uncorrelated with the ground truth motion field.

It would be possible to evaluate algorithm performance according to the definition above by measuring how well the optical flow field transforms one image into the next. However, this is inadequate for determining how well the algorithms work with 3D structure recovery applications. Horn and Schunck [8] state "One endeavors to recover an optical flow that is close to the motion field." A more appropriate measure then is the correlation between the calculated optical flow and the projected motion field.

A disadvantage of using this approach is that some measured errors will be due to the fundamental irrecoverability of the motion field from the given data rather than specific flaws in the tested algorithms. Also, it should be noted that this measure is better suited to applications that attempt to extract a three-dimensional structure, rather than video coding applications, since the latter need only be able to transform one image to another irrespective of the real motion field.

3. GROUND TRUTH MOTION FIELDS FOR SYNTHETIC SCENES

Although evaluations using real data sets are necessary, it is extremely difficult to obtain ground truth motion fields from real image sequences except in very limited or controlled circumstances. Nevertheless, it is useful to have access to more unconstrained sequences to provide a richer evaluation set. Further, a system that allows relatively easy generation of complex sequences and ground truth motion fields allows us to investigate how well algorithms work on different motion modalities. The following section contains a description of how the motion fields are generated, and Section 3.2 discusses the actual sequences we have created.

3.1. Generating Sequences and Motion Fields

The synthetic ground truth motion fields were generated using the publicly available tool presented by Mason *et al.* [12]. The motion field is generated during the ray tracing of two consecutive frames. The entire process is illustrated in Fig. 1. Rendering of the first frame begins by casting rays from the camera into the scene. Whenever a ray strikes an object, the 3D position of the intersection, the object, and the corresponding pixel position (x, y) are recorded. Before the second frame is rendered, a transformation is applied to each object. The transformation is also applied to each recorded ray–object intersection. Once this has been complete, rays are cast from each new intersection point to the camera. Geometry yields the new pixel position (x', y') and then subtraction yields the ground truth motion vector.

3.2. Generating Data Sets

To generate the synthetic data sets, we identified what we believe are three fundamental properties of image sequences: scene complexity, object motion, and camera motion. We have performed a three-factor factorial experiment along these three qualitative axes. We

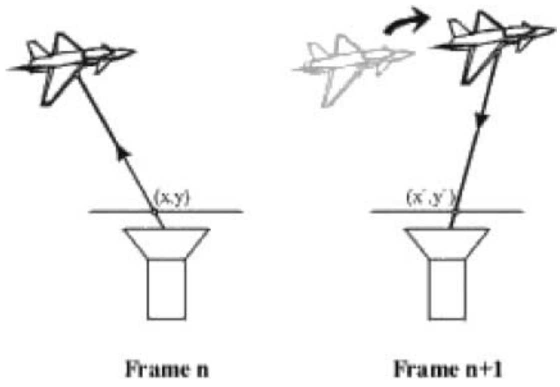


FIG. 1. Geometry for calculating the ground truth motion field.

have used four levels for object and camera motion (none, simple, medium, complex) and three for scene complexity (simple, medium, complex), resulting in a $3 \times 4 \times 4$ factorial experiment. We have decided to use only an exploratory data analysis as we do not believe that more formal statistical methods will contribute to the outcomes.

Scene complexity was varied by subjectively increasing the complexity of the objects in the scene. The simplest scene consists of a sinusoidally textured sphere and a noise-textured cube moving against a textured surface. A typical frame from this scene is shown in Fig. 2a. The medium complexity scene shows a single Volkswagen and a satellite moving against the same textured surface. This scene was considered more complex because the objects were not simple and were not smoothly textured. A typical frame is shown in Fig. 2b. A busy intersection was the subject of the most complex scene. This involves three Volkswagens driving through an intersection surrounded by many buildings, houses, and trees. A typical frame is shown in Fig. 2c.

For the camera and object motion, simple motion involved constant translation, medium motion involved constant rotation about a fixed point, and complex motion involved constant translation and constant rotation.

Clearly, this is not an exhaustive list for the complexity of image sequences. We have not tested for such things as moving light sources and oscillating motions. However, we believe these scenes do give an indication of the performance of the algorithms. Scene complexity should demonstrate how well an algorithm scales to more complex scenes that, at the most complex, include shadows, occlusions, and some light interactions between objects. The complexity of object motion should demonstrate if an algorithm can successfully estimate strong motions in small areas, and the complexity of camera motion should demonstrate if an algorithm can estimate global flows across an image.

4. GROUND TRUTH MOTION FIELDS FOR REAL SCENES

Testing the algorithms on synthetic scenes allows us to compare performance, but it gives us only a limited idea of how the algorithms will perform on real scenes. Since the representation and storage of highly detailed models and the accurate simulation of all physical lighting effects are beyond the capability of today's computer graphics systems, the only true test of computer vision algorithms remains their performance on real data.

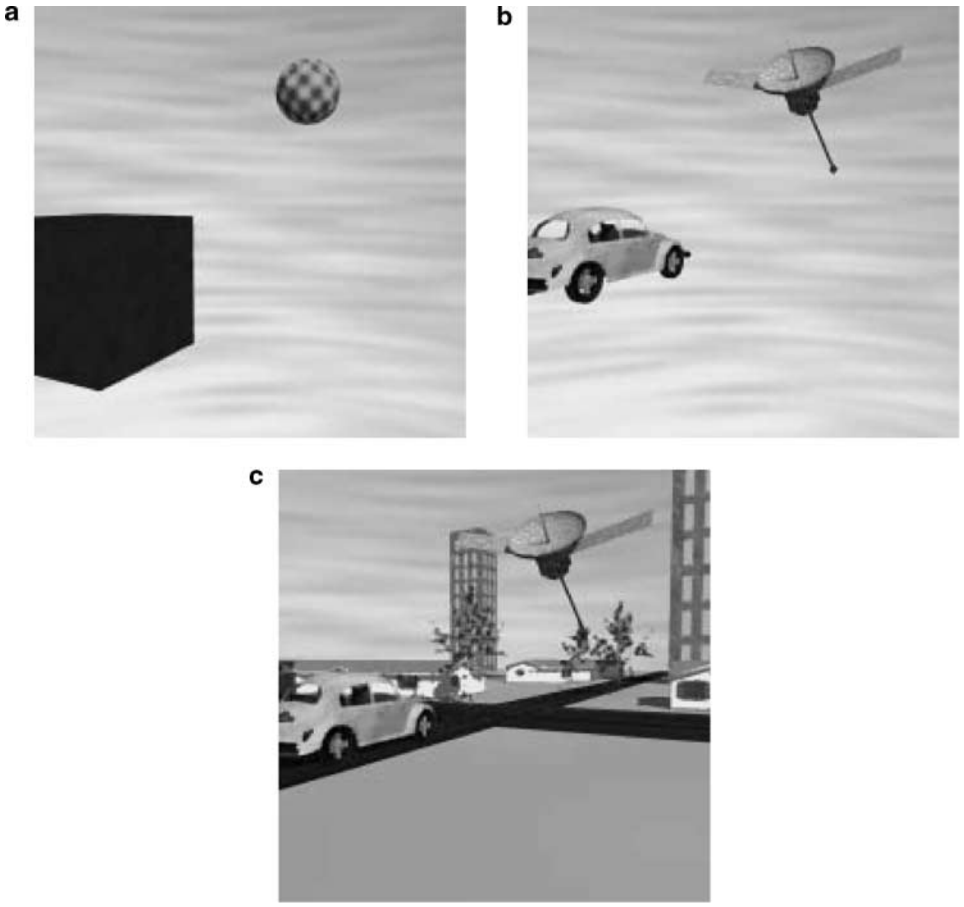


FIG. 2. Typical scenes from the test sequences. All image sequences were rendered at 400 $\times$ 300 resolution in full color and converted to 256 level gray-scale. We used 30 frames of each sequence for testing all of the algorithms. (a) Simple scene, (b) medium scene, and (c) complex scene.

Unfortunately, obtaining ground truth motion fields for arbitrarily complex image sequences requires an exact solution for the correspondence problem at every point in all the images. This is only possible in a limited set of circumstances. For example, Otte and Nagel [16] were able to generate the ground truth optical flow of a real sequence by mounting the camera on a calibrated robot arm and orienting it toward a measured block world environment.

We have developed a projective geometry technique for generating ground truth optical flow for real sequences of scenes containing planar polyhedral objects in front of a planar background. Only the background may intersect the image edges and the polyhedra must be unoccluded. Furthermore, all visible faces of the polyhedra must have at least four vertices, and the set of visible polyhedra faces may not change over the image sequence.

Once a sequence that fulfills these constraints has been captured, the user is required to manually identify and correspond all visible vertices of the polyhedra between each pair of frames. The vertices for each polyhedral face must be identified in clockwise or anti-clockwise order. Furthermore, at least four features on the background plane must be corresponded between each pair of images. Once these correspondences are performed, the ground truth motion field for the vertices in the scene is trivially available.

We then use the following result from projective geometry: Four points in two perspective images are sufficient to define a unique planar projective warp between the images [21]. This projective warp may be computed by solving a linear system of equations, which will be overdetermined if more than four points are available. Details are given in the Appendix.

Since each of our polyhedral faces has at least four vertices, we can always solve for the warp that corresponds all points on all faces between the two images. Since we have a clockwise ordering of the vertices of a face, we can easily determine the set of visible points that lie on each face. Similarly, we can define a warp for the planar background and use the resulting background correspondences for any points that lie outside the perimeter of all polyhedral faces.

This approach assumes that the images are perfect perspective transforms of the scene, which is unlikely to be the case near the perimeter of the image due to lens distortion [19].

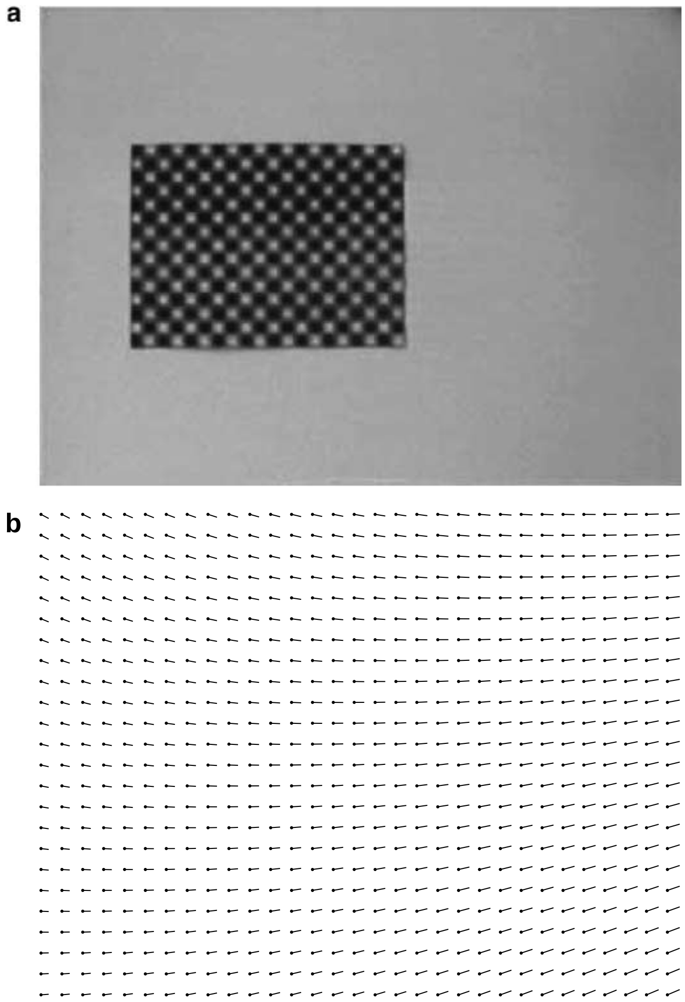


FIG. 3. A frame from the SineGrating sequence with its corresponding motion field. Every 10th vector is shown and the vectors are not scaled. (a) First frame and (b) corresponding motion field.

For this reason, camera calibration is highly recommended. The approach also suffers from numerical instability if the four points are not well separated along any axis. This is particularly important for the background quadrilateral, since the projective warp is extrapolated outside the quadrilateral as well as interpolated inside.

Although this method does not allow us to compute ground truth for real sequences that are significantly more complex than those computed by others, we feel that establishing a small number of manual correspondences in the images is much simpler than precisely measuring distances in the real world.

We have generated ground truth for three scenes using this technique. The first involves the camera panning across a simple two-dimensional sinusoid grating on a plain background (Fig. 3). The second involves zooming out from a view of a single cardboard box (Fig. 4), and the third involves a camera rotating around a number of toy blocks (Fig. 5). In each case, the scene is static and the hand-held camera is in motion. It would be possible to use the technique for measuring the flow from objects in motion, but this would

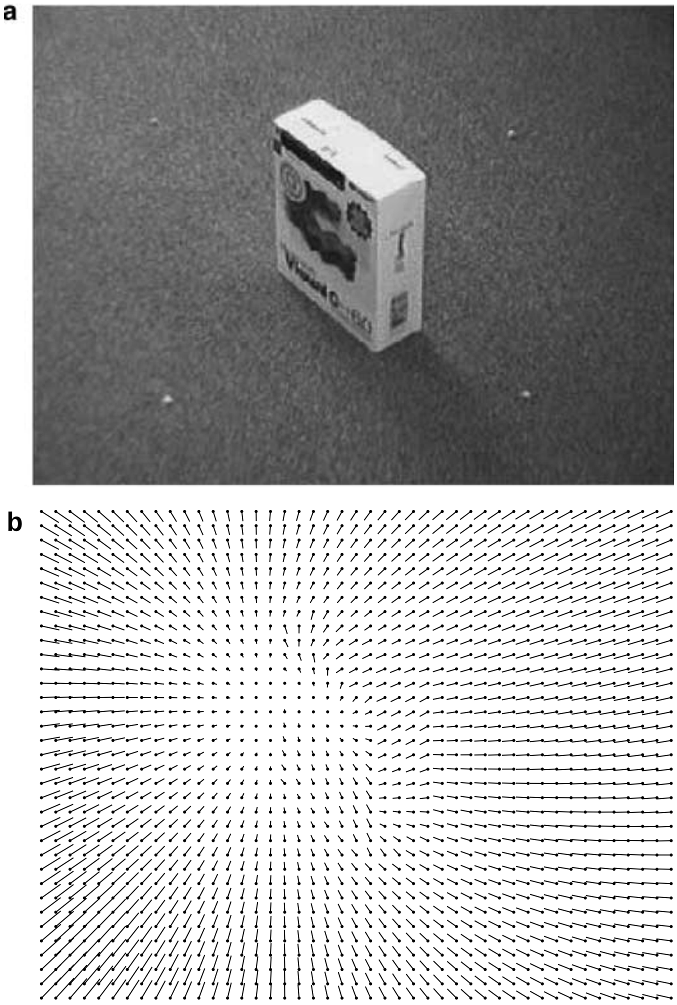


FIG. 4. A frame from the ZoomBox sequence with its corresponding motion field. Every 7th flow vector is shown and the vectors are scaled by a factor of 4. (a) First frame and (b) corresponding motion field.

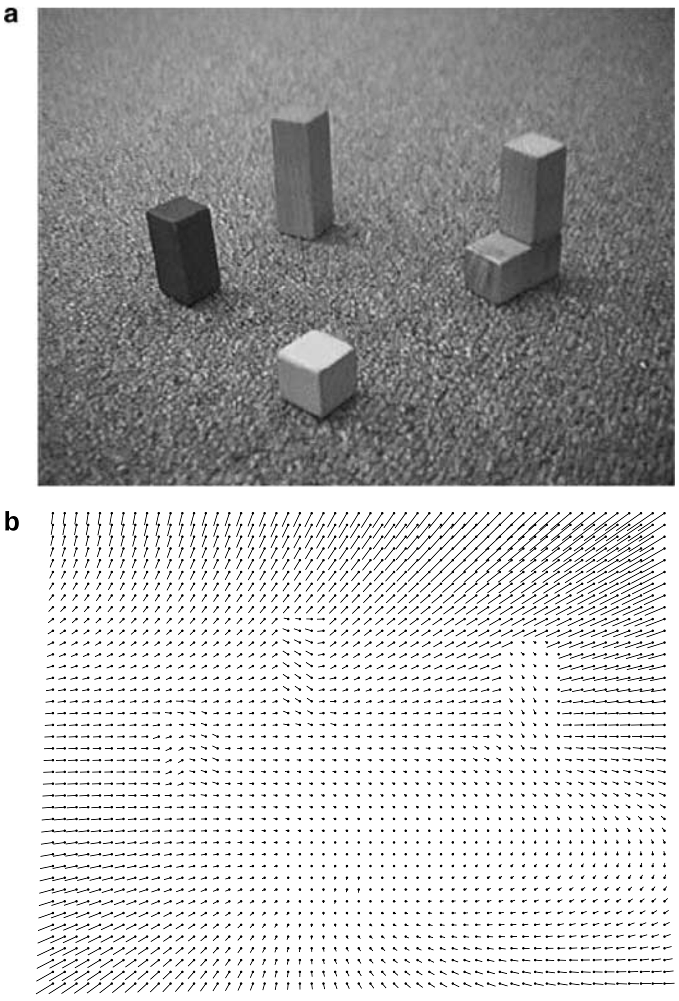


FIG. 5. A frame from the MoreBlocks sequence with its corresponding motion field. Every 7th flow vector is shown and the vectors are not scaled. (a) First frame and (b) corresponding motion field.

typically require the use of stop-frame animation, and would risk other difficulties such as occlusion or a change in the set of visible polyhedral faces. The technique could be extended to handle some partial occlusions by extrapolating lines and inferring corner positions.

5. ERROR METRICS

Trying to summarize the performance of over a million flow vectors with a single number is a difficult, if not impossible, task. It is further complicated by attempting to define a general, nonapplication specific measure. Otte and Nagel [16] criticized the angular error measure of Barron *et al.* [5] due to its bias for large flow vectors and its asymmetry. For this reason we decided not to use Barron *et al.*'s measure. Otte and Nagel [16], on the other hand, opted for simplicity and decided to use a simple magnitude of difference between estimated and correct flow. This is a reasonable measure, but in some ways it is insufficient because it effectively discounts errors in regions of small flow. We have decided to use two separate metrics to overcome some of these problems.

Our first error metric is the difference in angle between the correct and estimated flow vectors. We define

$$E_A = \cos^{-1}(\hat{c} \cdot \hat{e}), \tag{1}$$

where E_A is the error measure, c is the correct motion vector, e is the estimate optical flow vector, and ˆ denotes vector normalization. To avoid the problems of small flow vectors here, we have added a small number, δ, as a third coordinate to c and e. The effect of the third coordinate is larger for small flows than for large flows. For flows whose magnitude is less than or equal to δ, the angle difference is significantly lowered by the third coordinate, whereas it is largely unaffected for large flows. This is exactly the effect we wanted as we consider an incorrect angle to be more significant in areas of large flow than in areas of small flow (especially in areas of flow near 0). The magnitude of δ controls the threshold between small flows and large ones—for our experiments, we have used $\delta = 1$.

The second metric is the normalized magnitude of the vector difference between the correct and estimated flow vectors. The normalization factor is the magnitude of the correct flow vector. Again we need to account for small flows, which we do by choosing a threshold T that can be interpreted as a significance threshold. That is, we don't expect the algorithms to reliably produce accurate flow vectors in areas where the actual flow magnitude is less than T,

$$E_{\mathrm{M}} = \begin{cases} \dfrac{\|c - e\|}{\|c\|} & \text{if } \|c\| >= T, \\ \left|\dfrac{\|e\| - T}{T}\right| & \text{if } \|c\| < T \quad \text{and} \quad \|e\| >= T, \\ 0 & \text{if } \|c\| < T \quad \text{and} \quad \|e\| < T, \end{cases} \tag{2}$$

where E_{M} is the error measure. In all of the experiments performed here, $T = 0.5$ pixels. The perfect estimate would result in a normalized error score of 0.0, and a zero estimate would result in a normalized error score of 1.0. Scores larger than 1.0 indicate that the errors are larger than the real flows in the scene.

6. TESTING AGAINST STANDARD OPTICAL FLOW ALGORITHMS

Many of the algorithms tested here are implementations made available by Barron *et al.*, via anonymous ftp from `ftp.csd.uwo.ca` in the directory `/pub/vision`. These algorithms include:

- the coarse-to-fine area matching strategy of Anandan [3, 4],
- a modified version of Horn and Schunk's classic algorithm that uses temporal and spatial smoothing [9],
- the least-squares method of Lucas and Kanade and others [2, 10, 11, 18],
- the second-order derivative method of Nagel [13–15], and
- the local second-order differential method of Uras *et al.* [20].

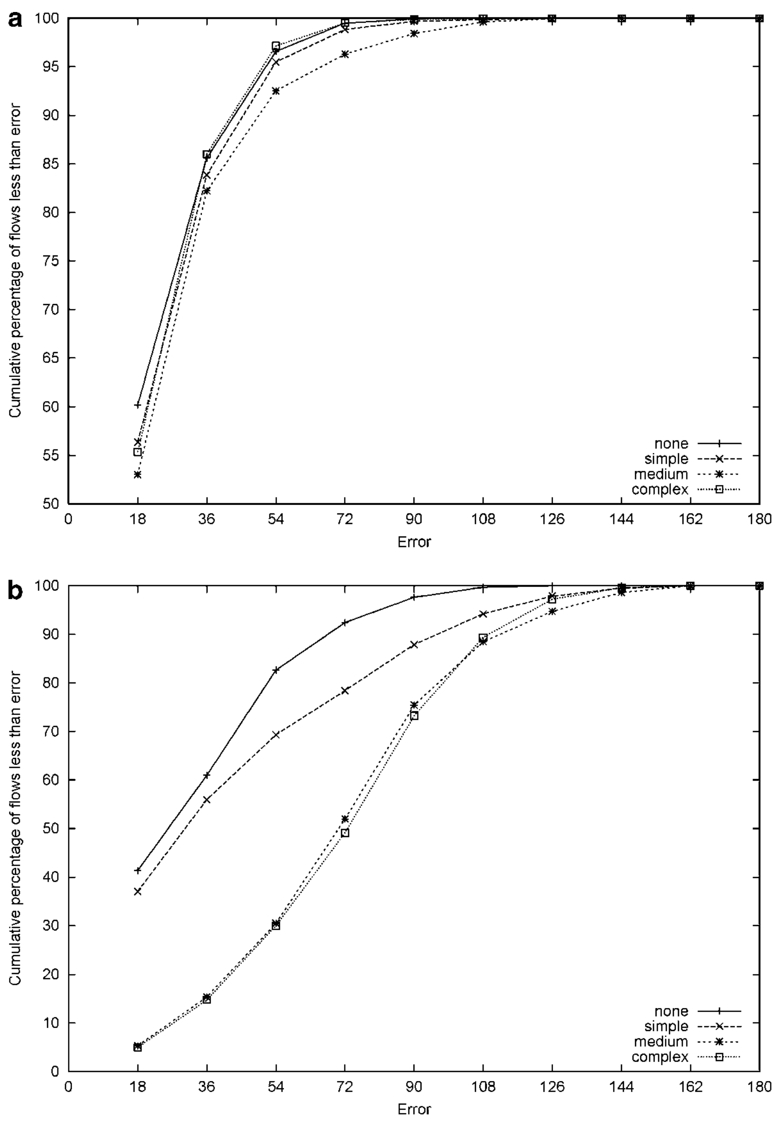


FIG. 6. Angle performance of (a) Proesmans' and (b) Nagel's algorithms with different levels of camera motion.

We have also evaluated two more recent algorithms:

- a multiscale version of the anisotropic diffusion method of Proesmans *et al.* [17], and
- the multiframe area matching method of Camus [6] using three frames.

Implementations for these latter two algorithms are provided on our web page [1]. For convenience, in the remainder of this paper we refer to the algorithms by the surname of the first author only.

The evaluations of these algorithms are provided for illustrative purposes only. We make no claim to the optimality of the implementations nor even to the optimality of the parameters we used (which were typically the default ones). We consider that it is extremely difficult

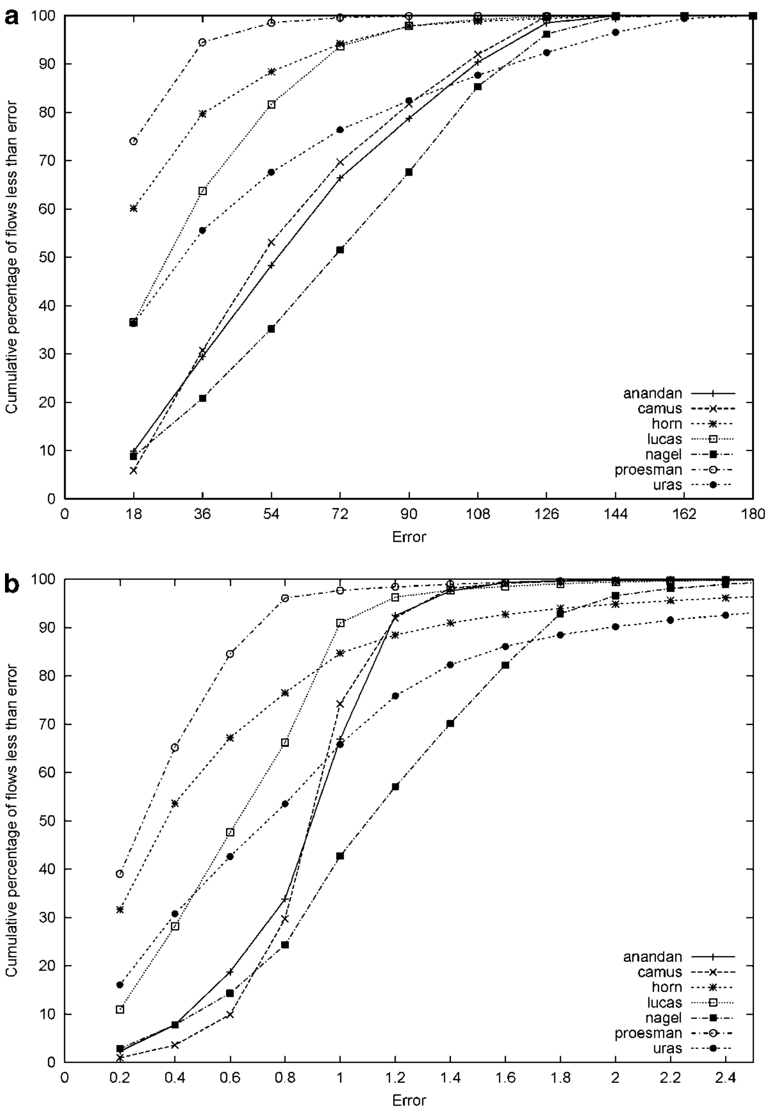


FIG. 7. Relative performance of each algorithm on the most complex synthetic scene. (a) Angle performance and (b) magnitude of difference performance.

to perform unbiased evaluations of implementations of other researchers' algorithms. We would rather see other researchers make their code available and test their algorithms themselves using our sequences. We hope that our sequences will form the core of an optical flow benchmarking suite that other researchers will make use of for their evaluations.

7. RESULTS

Here we display the results using both the magnitude of difference and angular error as discussed in Section 5. In both cases, we present the results in the form of cumulative histogram graphs that show the percentage of computed flow vectors within a given error

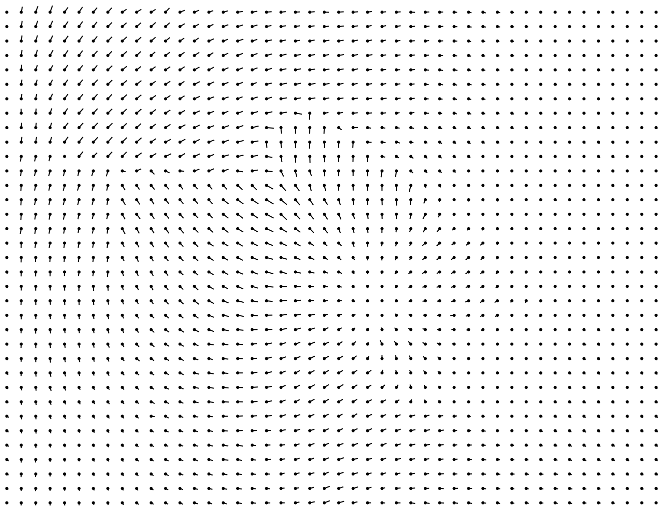


FIG. 8. The resultant flow map with Proesmans' algorithm on the ZoomBox sequence. Every 7th flow vector is shown and the vectors are scaled by a factor of 4. This map should be compared to the true projected motion field displayed in Fig. 4.

tolerance. Formally, a cumulative historgram is one in which the following is true,

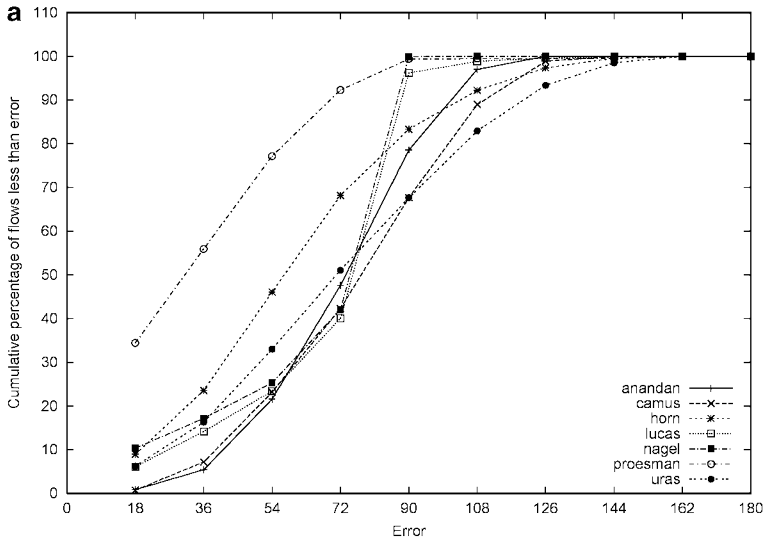
$$f(x) = \frac{\|P\|}{\|I\|}, \text{ s.t. } \forall p \in P, E_p \leq x, \tag{3}$$

where x is the error variable, I is the entire image, $P \subseteq I$, and E_p is the error at a particular pixel. For the magnitude of difference error, the histogram buckets are 0.2 wide and for the angular error, the histogram buckets are 18° wide. The ideal curve follows the left and top sides of the histogram, and for any particular error tolerance, the best algorithm is at the top of the graph.

7.1. Synthetic Sequences

We found that the synthetic scenes produced faithful rankings of the algorithms compared to the real scenes even though the raw data was not comparable. Figure 7 shows the results of each of the algorithms for both the angle and magnitude of difference error measures for each of the algorithms on the most complex synthetic sequence (complex camera motion, object motion, and scene complexity). The relative orderings in Fig. 7 are consistent with the results for the real sequences shown in Figs. 9, 10 and 11. The results were similar for the other synthetic scenes, so these are not included, although the data are available on our web page [1].

We also found that the evaluation of the synthetic scenes was useful for testing how well algorithms scaled to more complex sequences. In particular, we found that camera motion produced the most interesting results, with some algorithms scaling poorly with increased camera motion complexity, and some algorithms scaling well. This is displayed in Fig. 6, which shows that Proesmans' algorithm performs similarly with different levels of camera motion, while Nagel's algorithm performs much worse for motion involving camera rotation.



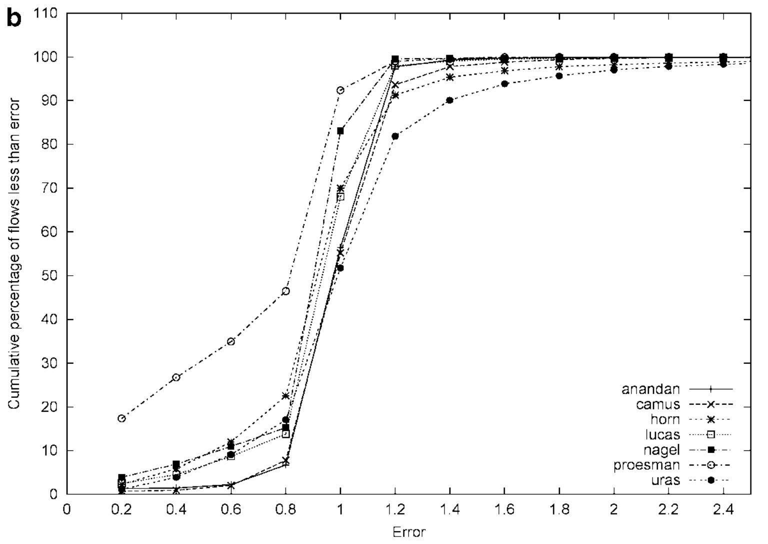


FIG. 9. Performance of the different algorithms with the Sinusoid grid sequence. (a) Angle results and (b) magnitude of difference results.

7.2. Real Sequences

We have also tested each of the algorithms on five frames from the three real sequences described above in Section 4. The results for each of the algorithms with each of the error metrics are displayed in Figs. 9, 10, and 11. Figure 8 shows the calculated flow from Proesmans' algorithm for the ZoomBox sequence. This is the same frame as the real flow displayed in Fig. 4. In the absence of ground truth data, one might be tempted to say that Proesmans' algorithm performs well; however, quantitatively it has not. This clearly demonstrates the problems with performing purely qualitative comparisons.

The magnitude of the average correct flow vector for these scenes are approximately 3.5 pixels for the ZoomBox sequence, 5.8 pixels for the MoreBlocks sequence and 6.7 pixels for the SineGrating sequence. Our multiscale implementation of Proesmans' algorithm

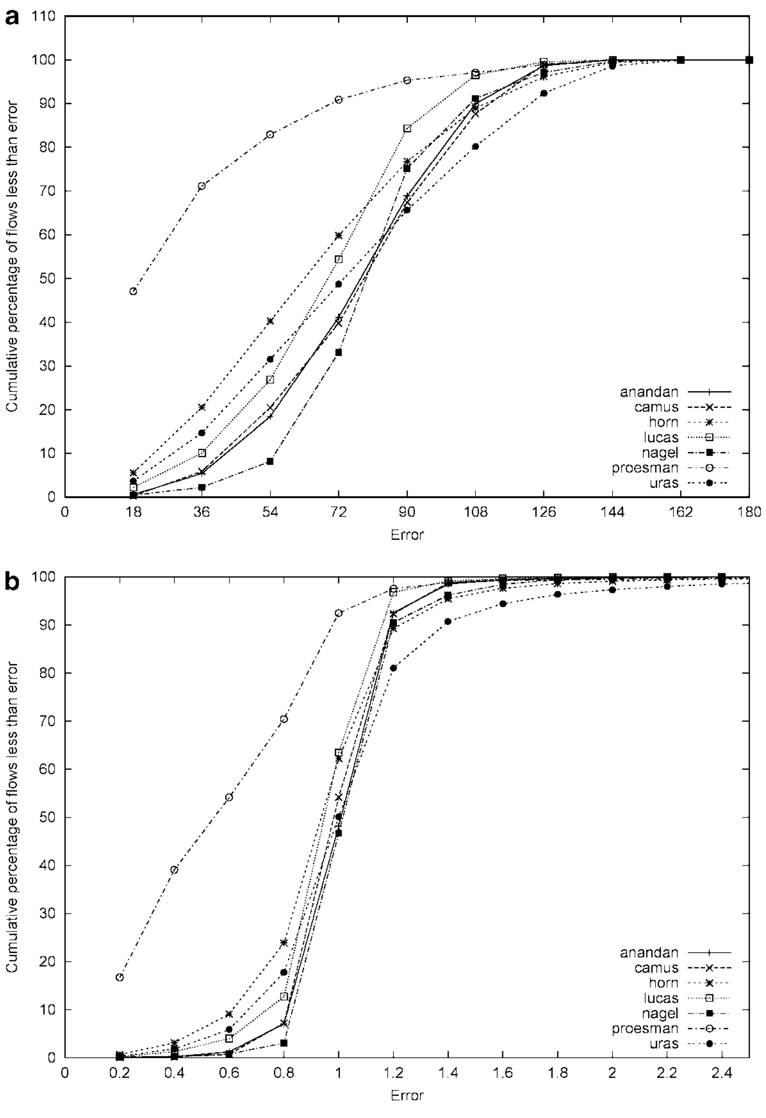
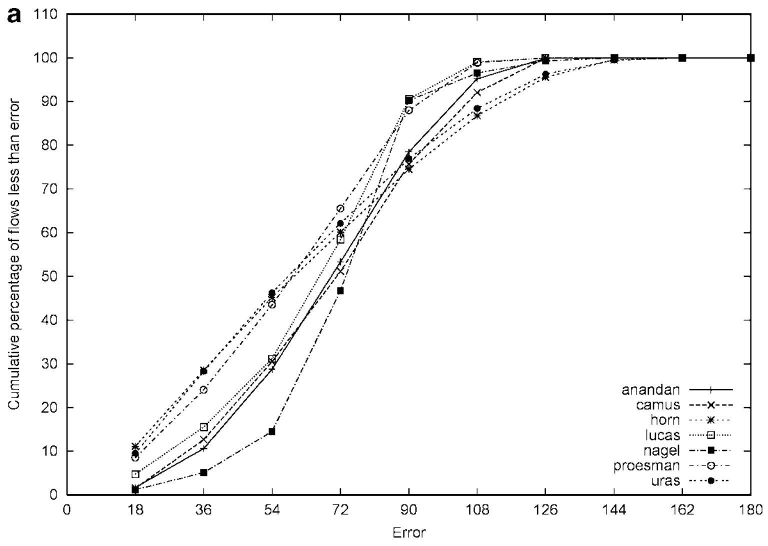


FIG. 10. Performance of the different algorithms with the toy blocks sequence. (a) Angle results and (b) magnitude of difference results.

is the stand-out performer—perhaps due to its multiscale nature. However, we initially implemented Proesmans' algorithm for a practical project. Hence, it was carefully optimized for accuracy. In contrast, we did not implement or optimize the algorithms studied by Barron *et al.* [5]. For the purposes of benchmarking, it is important that the algorithms be optimized and tuned by people who fully understand the algorithms and are motivated to produce the best possible result.

In a relative sense, the algorithms perform similarly on both real and synthetic sequences. It is interesting to note that our rankings differ from those of Barron *et al.* [5]. The likely cause of the disparity is that they have thresholded low confidence flows and we have not.



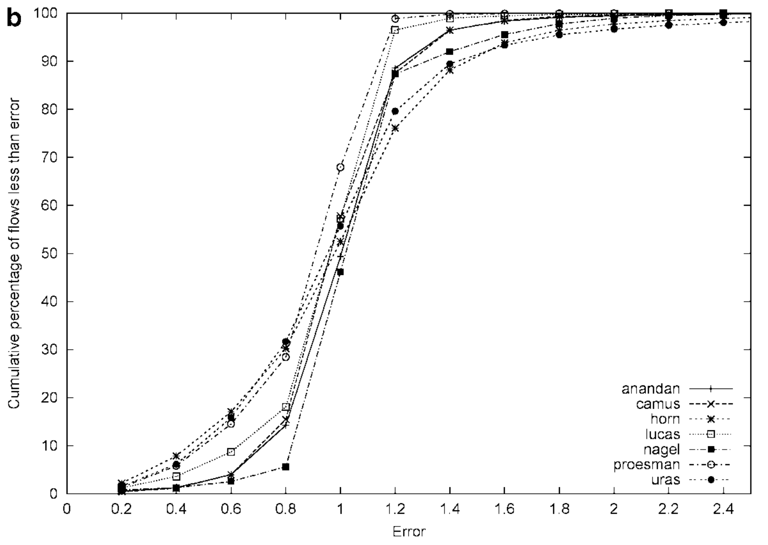


FIG. 11. Performance of the different algorithms with the ZoomBox sequence. (a) Angle results and (b) magnitude of difference results.

8. CONCLUSION

In this paper, we have introduced a comprehensive test set of synthetic sequences for evaluating optical flow algorithms. We have also developed a method for measuring the scene motion from real scenes containing polyhedral objects. This has allowed us to objectively evaluate the performance of several optical flow algorithms. The results of the synthetic and real scenes were consistent in a relative sense, thus verifying the validity of performing quantitative evaluations using synthetic data.

In general, the results for the real sequences are disappointing. The real question is: Are they useful for real applications? Clearly, the algorithm that performs the best in a quantitative sense should be the most useful for real applications, and therefore quantitative evaluation must be the yardstick against which algorithms are measured. Nevertheless, real applications and higher level processes may not require perfect results to perform according to specifications. Therefore, application-oriented evaluations (not qualitative) should also be applied to determine the usefulness of an algorithm in a certain context.

We do not believe that large comparative studies such as that presented in Barron *et al.* [5], or that presented here, are the ideal methods of performing evaluations of algorithms. This is largely due to the difficulty of faithfully reimplementing someone else's algorithm from a text description. Rather, we believe to improve empirical evaluation of optical flow and other computer vision areas, researchers in the field need to do the following:

- Make their code publically available.
- Evaluate their algorithms on a standard benchmarking suite of scenes using standard error metrics.
- Compare their results with those maintained at a central repository.

We hope that our work has gone some way to achieving this aim and are willing to maintain our site as a central repository of results and sequences.

9. FUTURE WORK

Future work will largely include maintaining and updating our web site as submissions are made to it. We hope that the submissions will include ground truth for a larger set of real scenes as this will strengthen any evaluations performed. There is also potential for a more careful study of the ability of algorithms to work under different conditions. Our work on synthetic sequences was a first attempt at such a study, but it has a limited number of factors that are studied, and although we tried to be thorough, the choice of complexities was necessarily arbitrary. The study of more factors (such as different lighting conditions, pan, tilt and zoom camera motions) and a more formal definition of complexity would be useful.

APPENDIX

The Projective Warp

As described in Section 4, the manual correspondence of the projections of four coplanar points in two perspective images defines a perspective warp between those two images. This warp can be used to automatically correspond the projections of all remaining points in the same world-space plane. This Appendix describes how to compute the perspective warp between the images given the correspondence. We follow the development given by Wolberg [21]. The reader is referred to his book for details, particularly if more than four manual correspondences are supplied.

Let (x, y) be a point in the first image and (x', y') be a point in the second image. A planar perspective warp between these images is based on eight parameters, which we call $a_1, a_2, \ldots, a_8$:

$$x = \frac{a_1 x' + a_2 y' + a_3}{a_7 x' + a_8 y' + 1} \tag{A.1}$$

$$y = \frac{a_4 x' + a_5 y' + a_6}{a_7 x' + a_8 y' + 1}. \tag{A.2}$$

In order to solve for the eight unknown parameters, we need eight equations. Each known correspondence between the two images supplies two equations: one for x and one for y. Therefore, four correspondences are sufficient to define the warp. Let the four correspondences be

$$\begin{aligned}(x_1, y_1) &\leftrightarrow (x_1', y_1')\\(x_2, y_2) &\leftrightarrow (x_2', y_2')\\(x_3, y_3) &\leftrightarrow (x_3', y_3')\\(x_4, y_4) &\leftrightarrow (x_4', y_4').\end{aligned}$$

Substituting these known values into Eqs. (A.1) and (A.2) yields eight equations. Rearranging these into matrix form gives the 8×8 linear system

$$\begin{bmatrix} x_1' & y_1' & 1 & 0 & 0 & 0 & -x_1'x_1 & -y_1'x_1 \\ x_2' & y_2' & 1 & 0 & 0 & 0 & -x_2'x_2 & -y_2'x_2 \\ x_3' & y_3' & 1 & 0 & 0 & 0 & -x_3'x_3 & -y_3'x_3 \\ x_4' & y_4' & 1 & 0 & 0 & 0 & -x_4'x_4 & -y_4'x_4 \\ 0 & 0 & 0 & x_1' & y_1' & 1 & -x_1'y_1 & -y_1'y_1 \\ 0 & 0 & 0 & x_2' & y_2' & 1 & -x_2'y_2 & -y_2'y_2 \\ 0 & 0 & 0 & x_3' & y_3' & 1 & -x_3'y_3 & -y_3'y_3 \\ 0 & 0 & 0 & x_4' & y_4' & 1 & -x_4'y_4 & -y_4'y_4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}, \tag{A.3}$$

where only $a_1, a_2, \ldots, a_8$ are unknown. This linear system can be solved using a number of different algorithms [7], for example, the singular value decomposition.

Once the values for $a_1, a_2, \ldots, a_8$ are established, the point (x, y) in the first image corresponding to a point (x', y') in the second image can be computed using Eqs. (A.1) and (A.2).

REFERENCES

1. University of Otago, Computer Vision Homepage, available at `http://www.cs.otago.ac.nz/research/vision/index.html`.
2. E. Adelson and J. Bergen, Spatiotemporal energy models for the perception of motion, in *Proc. IEEE Workshop on Visual Motion*, 1986, pp. 151–156.
3. P. Anandan, *Measuring Visual Motion from Image Sequences*, Ph.D. thesis, University of Massachusetts, 1987.
4. P. Anandan, A computational framework and an algorithm for the measurement of visual motion, *Internat. J. Comput. Vision* **2**, 1989, 283–310.
5. J. Barron, D. Fleet, and S. Beauchemin, Performance of optical flow techniques, *Internat. J. Comput. Vision* **12**, 1994, 43–77.
6. T. Camus, Real-time quantized optical flow, *J. Real-Time Imaging* **3**, 1997, 71–86.
7. G. H. Golub and C. F. V. Loan, *Matrix Computations*, 2nd ed., Johns Hopkins Univ. Press, Baltimore, MD, 1989.
8. B. K. P. Horn and B. Schunck, Determining optical flow: A retrospective, *Artif. Intell.* **59**, 1993, 81–87.
9. B. K. P. Horn and B. G. Schunck, Determining optical flow, AI Memo 572, Massachusetts Institute of Technology, 1980.

10. J. Kearney, W. Thompson, and D. Boley, Optical flow estimation: An error analysis of gradient based methods with local optimization, *IEEE Trans. Pattern Anal. Mach. Intell*. **9**, 1987, 229–244.
11. B. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, in *Proc. DARPA Image Understanding Workshop*, 1984, pp. 121–130.
12. D. Mason, B. McCane, and K. Novins, Generating motion fields for the evaluation of optical flow algorithms on complex scenes, in *Computer Graphics International (CGI), Canmore, Canada, 1999*, pp. 65–69. IEEE Computer Society Press, Los Alamitos, CA.
13. H. Nagel, Displacement vectors derived from second-order intensity variations in image sequences, *Comput. Graphics Image Process*. **21**, 1983, 85–117.
14. H. Nagel, On the estimation of optical flow: Relations between different approaches and some new results, *Artif. Intell*. **33**, 1987, 299–324.
15. H. Nagel and W. Enkelmann, An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences, *IEEE Trans. Pattern Anal. Mach. Intell*. **8**, 1986, 565–593.
16. M. Otte and H.-H. Nagel, Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-interlaced image sequences, *Artif. Intell*. **78**, 1995, 5–43.
17. M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck, Determination of optical flow and its discontinuities using non-linear diffusion, in *3rd European Conference on Computer Vision, ECCV'94*, 1994, Vol. 2, pp. 295–304.
18. E. Simoncelli, *Distributed Representation and Analysis of Visual Motion*, Ph.D. thesis, MIT, 1993.
19. R. Y. Tsai, A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses, *IEEE J. Robot. Autom*. **3**, 1987, 323–344.
20. S. Uras, F. Girosi, A. Verri, and V. Torre, A computational approach to motion perception, *Biol. Cybernet*. **60**, 1988, 79–97.
21. G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, Los Alamitos, CA, 1990.