Determination of Egomotion and Environmental Layout From Noisy Time-Varying Image Velocity In Binocular Image Sequences

John L. Barron, Allan D. Jepson* and John K. Tsotsos*

Department of Computer Science
University of Toronto
Toronto, Canada, M5S 1A4

Abstract

We address the problem of interpreting time-varying image velocity fields generated by a moving binocular observer viewing a stationary environment under perspective projection to obtain 3-D information about the absolute motion of the observer (egomotion) and the absolute depth of environmental surface points. We conduct a numerical study of our algorithm (which involves solving nonlinear systems of equations) for best, random and worst case image velocity error. As well, we investigate how good the initial guess for the nonlinear system of equations has to be. Other results include the presence of multiple solutions in time, how the algorithm performs when the underlying assumptions it is based on are violated and the effect of varying the spatial extent of the image points used, of varying the spatial baseline (separation of the left and right cameras) and of varying the temporal extents of the image points used (effectively varying the temporal baselines). As well, we investigate the use of convergent/divergent as opposed to parallel stereo camera setups.

1 Introduction

In this paper we present an algorithm for computing the motion and structure parameters that describe egomotion and environmental layout for a moving binocular observer viewing a stationary environment This algorithm is an extension of the monocular algorithm presented in [Barron ct al 87a]; the two coincide when the left and right image sequences coincide and the temporal baselines at each time are unknown. The binocular motion and structure parameters are simply the monocular motion and structure parameters,!/, the depth scaled ovulational observer velocity, , the observer's rotational velocity and a, the normalized surface gradient of some planar surface, plus*,, the absolute depth of some point on the surface.

Our algorithm reconstructs observer motion and environmental structure by solving a nonlinear system of equations; each equation relates image velocity at some image point, $\vec{Y}_{lr}(\vec{P}',t)$, in either the left or right image sequence to the underlying motion and structure parameters in the left image sequence at the solution point, $\vec{Y}_l(\vec{P},t)$.

In general, monocular and binocular reconstruction have been considered two separate problems. Monocular reconstruction typically involves solving systems of (nonlinear) equations relating image velocity (and possibly its 1st and 2nd order spatial/temporal derivatives) to the underlying motion and structure parameters describing a surface in relative motion with an observer (see [Barron 84]). The classical stereo paradigm proposes that 3-D depth be recovered by computing matching primitives in the left and right images of a stereo pair, establish correspondence between the appropriate primitives in the two images and then calculating 3-D depth using simple trigonometry. [Barnard and Fischler 82], [Jenkin 84] and [Poggio and Poggio 84] provide surveys of some of the current stereo techniques.

We believe that monocular and binocular vision have a lot in common and can be solved in a unified way. The algorithm presented in this paper is more in the favour of monocular reconstruction algorithms that interpret image velocity fields, for example [Longuet-Higgins and Prazdny 80] or [Waxman and Ullman 85] then in the favour of the classical stereo paradigm described above. We also interpret image velocities fields; but we do so by sampling the image velocities at many discrete times in both left and right stereo image sequences.

[Waxman and Duncan 86] and [Waxman and Wohn 86] have also proposed that left and right monocular image velocity fields can be analyzed to compute depth. Their algorithm involves the computation of relative flow (or binocular difference flow). As such stereo correspondence must still be computed. Some researchers, such as [Kanatani 85] and [Aloimonos and Rigoutsos 86], have advocated a correspondence-less approach for monocular reconstruction.

Only a few researchers have begun to address the use of temporal information, such as temporal derivatives, in reconstruction [Subbarao 86a], [Bandopadhay and Aloimonos 85]. We note that others' use of temporal derivative information and our use of time-varying image velocities are approximately equivalent; image velocity fields (at least locally) can be derived from one image velocity and its 1* and/or 2nd spatial and temporal derivatives and vice-versa. Indeed, image velocity fields are often used in the derivation of spatial and temporal image velocity information [Waxman and Wohn 85].

There has been little or no error analysis in previous monocular reconstruction work. Some researchers, such as [Waxman and Ullman 85], [Buxton et al 84], [Aloimonos and Rigoutsos 86], [Snyder 86] and [Subbarao 86b] have begun to consider the inherent insensitivity of their algorithms for noisy input. Their reports usually consist of a few runs of their algorithms with random noise in the input.

1.2 Underlying Assumptions

In order to relate a stereo spatio-temporal distribution of image velocity to the motion and structure parameters at some image point, $\vec{Y}_{lr}(\vec{P},t)$, we make 5 assumptions:

(a) 3-D objects in the environment are rigid. The rigidity assumption ensures that the image velocity of an object's point is due entirely to the point's motion with respect to the observer and not due to changes in the object's shape.

(b) 3-D surfaces are planar. This local planarity assumption means curved surfaces can treated as collections of adjacent planes.

(c) The observer rotates with a constant angular velocity for some small time interval. This is called the fixed axis assumption [Webb and Aggarwal81].

(d) All image velocities used in a particular calculation of motion and structure are measured with respect to the same 3-D planar surface. We call this the same surface assumption.

(e) The observer's translational velocity is constant with respect to the scene frame of reference.

These assumptions allow us to design our algorithm so that we do not have to solve point-to-point correspondence either in individual left and right image sequences or between stereo images (1).

2 Algorithm Description

In this section, we present a brief description of our algorithm. Complete details are given in [Barron 87].

2.1 Notation

We use notation $\vec{P}_l(t;\tau)$ and $\vec{P}_r(t;\tau)$ to indicate a 3-D point measured at time t in left and right coordinate systems, $\vec{X}_l(\tau)$ and $\vec{X}_r(\tau)$, respectively. We use subscript lr to refer to quantities that can be in either the left or right coordinate systems. Equations with quantities subscripted lr are interpreted in a consistent way; all quantities are either l or r but not some mixture. Thus, $X_3(\vec{P}_{lr},t;\tau)$ is the depth of $\vec{P}_{lr}(t;\tau)$ in $\vec{X}_{lr}(\tau)$ coordinates. $\vec{Y}_{lr}(\vec{P}_{lr},t)$ is the image of $\vec{P}_{lr}(t;t)$.

(1) Of course we must still solve surface correspondence, i.e. group spatio-temporal distributions of image velocity that belong to the same planar surface. See [Adiv 84] for one approach.

* Alio, Canadian Instute for Advanced Research

2.2 Physical Setup

We model the left and right observation points using right hand coordinate systems as in Longuet-Higgins and Prazdny [80]. $\vec{U}_r=(U_1,U_2,U_3)$ is the translational velocity of the observer centered at the origin of the coordinate system $\vec{X}_r(t)$. $(0,0,X_3)$ is the line of sight in $\vec{X}_r(t)$. $\vec{\omega}_r=(\omega_1,\omega_2,\omega_3)$ is the angular velocity of the observer. $\vec{X}_l(t)$ and $\vec{X}_r(t)$ are rigidly connected, hence, $\vec{\omega}_l=\vec{\omega}_r=\vec{\omega}$ The center of the two coordinate systems, the left and right observation points, are separated by a spatial baseline $\vec{s}(t;t)=(s_1,s_2,s_3)$ and $\vec{X}_r(t)$ is rotated with respect to $\vec{X}_l(t)$ by $R_E(\phi_1,\phi_2,\phi_3)$; ϕ_1,ϕ_2 and ϕ_3 are Euler angles as defined by Arfken [70, pp178-180] for a right-hand coordinate system. If $(\phi_1,\phi_2,\phi_3)=(0,0,0)$ the left and right coordinate systems are parallel; otherwise we have either a convergent or a divergent stereo setup. In addition to a spatial baseline, the observation point in either a left or right sequence at $\vec{X}_b(t)$ can be separated by a temporal baseline $\vec{T}_b(t,t';t)$ from $\vec{X}_b(t')$. $\vec{X}_l(0)$ is the inertial coordinate system.

2.3 The General Image Velocity Equation

We can write an equation relating image velocity at a point $\vec{Y}_r(\vec{P}_{rk},t')$ to the binocular motion and structure parameters at some point $\vec{Y}_l(\vec{P}_{lk},t)$ as

$$\vec{v}_r(\vec{Y}_r(\vec{P}_{rk},t'),t')=A_1(\vec{Y}_r(\vec{P}_{rk},t'))h(\vec{Y}_l(\vec{P}_{lk},t))R_E^T(\phi_1,\phi_2,\phi_3)$$

$$\left[Q_R(\vec{\omega},t,t')\vec{u}_l(\vec{Y}_l(\vec{P}_{lk},t),t;t)+\frac{\vec{\omega}(t;t)\times\vec{s}(t;t)}{h(\vec{Y}_l(\vec{P}_{lk},t))X_3(\vec{P}_{lk},t;t)}\right]$$

$$TST_B(\vec{Y}_l(\vec{P}_{lk},t),t,t',t')S_{rk}(\vec{Y}_l(\vec{P}_{lk},t),\vec{Y}_r(\vec{P}_{rk},t'),t')$$

$$+A_2(\vec{Y}_r(\vec{P}_{rk},t'))\vec{\omega}(t;t) \qquad (2.3\text{-}1)$$

where $\vec{P}_l$ and $\vec{P}_r$ are 3-D points on the same planar surface and generally $\vec{Y}_l(\vec{P}_{lk},t)\neq\vec{Y}_r(\vec{P}_{rk},t')$. In a left image sequence, $\vec{v}_r(\vec{Y}_r(\vec{P}_{rk},t'),t')$ reduces to $\vec{v}_l(\vec{P}_{lk},t'),t')$ provided we use $\vec{s}=(0,0,0)$, $R_E^T=I$ and note that $\vec{T}(t,t',t'';t)$, the spatio-temporal baseline, is simply $\vec{T}_l(t,t'';t)$. Complete details concerning this equation are given in [Barron 87].

The use of a spatio-temporal distribution of image velocities requires that we make assumptions about the kinds of motion the observer is undergoing. In this paper, we consider 2 specific types of motion, although we emphasize that our treatment can be generalized to other motions as well. The 2 types of motion considered are:

Type 1: A vehicle is moving with constant translational velocity and has a camera mounted on it that is rotating with constant angular velocity.

Type 2: A vehicle with a fixed mounted camera is moving with constant translational and angular velocity.

3 Experimental Technique

In this section we discuss the implementation of our algorithm and present the details of our error analysis.

3.1 Implementation

Newton's method is used to solve the systems of non-linear equations. Since only 2 components of $\vec{\alpha}$ are independent, we add an extra normalization row to the Jacobian matrix, J, to ensure the computed $\vec{\alpha}$ is normalized; hence J is a full rank 10 matrix. The 10^{th} value of $\vec{f}_M$, the measured image velocities is then set to 1.

When $\vec{\omega}$ is known to be zero, i.e. in the case of pure translation (type 1 and type 2 motions are equivalent here) we can use a 7×7 Jacobian instead of a 10×10 one. We compute a 10×7 Jacobian (the 3 columns corresponding to $\vec{\omega}$ are not computed). We let the LU decomposition of J choose the bests 7 rows of J, with the provision that the normalization row is always one of the chosen rows.

3.2 Error Analysis

We compute an error vector, $\vec{\Delta f}_M$ which, when added to $\vec{f}_M$, yields the perturbed input, i.e.

$$\vec{f}_M'=\vec{f}_M+\vec{\Delta f}_M. \qquad (3.2\text{-}1)$$

For X% random case error [2], we compute five random 2-component unit vectors, $\hat{n}_j$, $j=1,...,5$, and then compute each i^{th} component of Δf_M as

$$\begin{bmatrix}\Delta f_{iM}\\ \Delta f_{i+1M}\end{bmatrix}=\frac{X}{100}\hat{n}_j\,||\vec{v}_j||_2,\ j=1,...,5,\ i=j\times2-1. \qquad (3.2\text{-}2)$$

We use $\vec{v}_5$ in the calculation of error for v_{51}, in Δf_{9M}. Δf_{10M} is 0, i.e. we do not add error to the normalization constant. Using $\vec{\Delta f}_M$ for random error we compute $\Delta f_{norm}=||\vec{\Delta f}_M||_2$. We use forward and inverse iteration on J [3] to compute normalized best and worst case error directions, $\hat{e}_B$ and $\hat{e}_W$. We compute $\vec{\Delta f}_M=\hat{e}_B\,\Delta f_{norm}$ as X% best case scaled image velocity error and $\vec{\Delta f}_M=\hat{e}_W\,\Delta f_{norm}$ as X% worst case scaled image velocity error. Both best and worst $\vec{\Delta f}_M$ are made to be the same size as the random $\vec{\Delta f}_M$ for comparison purposes. A last type of error involves adding worst case error to the image velocities so that the maximum error in any image velocity is X%. We call this **worst case relative** error.

We compute initial guess error by adding X% random error individually to $\vec{u}$, $\vec{\alpha}$, $\vec{\omega}$ and X_3.

4 Experimental Results

In this section, we present some preliminary results obtained from testing our algorithm.

We consider 3 motions: (1) $\vec{U}_l=(0,0,1000)$, $\vec{\alpha}_l=(0,0,1)$, $\vec{\omega}=(0,0,0)$ and $X_{l3}=2000$, (2) $\vec{\omega}$ is changed to 0,0.1,0 for type 1 motion and (3) a>,=(0,0.l,0) for type 2 motion. The spatial baseline is=80,0,0) (measured at time 0) for a parallel setup, i.e. (0i.02.to)=(O,O,O). Motion 1 corresponds to pure observer translation towards a wall. Motions 2 and 3 correspond to an observer translating directly towards a wall as he rotates his head to the right (type 1 motion) or moving towards the wall while turning to the right (type 2 motion). These motions are singular if all image velocities are measured at time 0. However motion and structure can be recovered from a spatio-temporal distribution of image velocity. We measure 4 image velocities at points, (70,70) at time 0, (-30,70) at time r/4, (-30,-30) at time *til* and (70,-30) at time 3</4, i.e. at the four corners of a square centered at the solution point (20,20). Since we need 9 image velocities for our binocular algorithm, we measure the y, image velocity component at (20,20) at time *t*. The solution is computed for time 0. These image coordinates are assumed to be measured on a 256x256 display device and so are scaled by 256 to produce realistic /coordinates. Thus, (20,20) in pixels is scaled to (0.078125,0.78125) in / coordinates. The viewing angle of these points is computed as the maximum diagonal angle subtended by the points, i.e. 33.05°. We call this the spatial extent The temporal extent, written as 0-t, refers to the five times used, i.e. 0, *t*/4, r/2, 3//4 and / as above.

8x8 tables are used to display the output error for runs made by varying two quantities, say temporal extent against image velocity error. In this case table rows (from left to right) correspond to increasing temporal extent while table columns (from top to bottom) correspond to image velocity error.

Due to space limitations, we can only report a few of the results we have obtained to date. Tables 4-la,b,c,d,e,f show the output error for runs where temporal extent is varied from 0-0.3 to 0-1 in 8 equal steps and image velocity error is varied from 0% to 1.4% in 0.2% steps for random and worst case error directions for the three motions. The L_m condition numbers of the various Jacobian matrices are quite large; the values vary from 300,000 to 3,000,000! This indicates unstability. All 100% output error values correspond to unsolved runs; all other values, including those over 100%, correspond to solved runs. We do not show best case error results as these are effectively 0% (less than 0.2%). Indeed, even when maximum best case error was 49% the output error was only a few percent. These best case results are quite good, especially when compared with random and worst case output. The output error in the random cases is about 1/3 to 1/2 the output error in the worst case. As we can see, increasing the temporal extent significantly reduces output error, time appears to

(2) Computing random image velocity error in this way prevents the error that is added to the individual velocities from being too large relative to the magnitudes of the velocities, i.e. the error added to each image velocity pair depends on the magnitude of that image velocity. Since the magnitudes of the various velocities can vary greatly any technique for computing error that doesn't take this into account may end up adding very large error to the smaller image velocities.

(3) We note that the best and worst directions so calculated are for the initial linear system of equations, $J\vec{h}_0=\vec{f}_0$. It is possible that the actual best and wont directions for the nonlinear system of equations are different, although we expect these directions to be quite close to the computed best and wont directions.

increase robustness for these motions. The results indicate that worst case error of as little as 1.4% can produce unusable output, if we assume only output error that is less than 10%-20% is useful. It seems we need image velocity measurements to be quite accurate.

The second experiment involves using perfect image velocity data and varying temporal extent from 0–0.3 to 0–1 as before while varying initial guess error from 0% to 100% in 8 equal steps for the 3 motions. All 100% output error values indicate unsolved runs while 0% output error indicates solved runs. For the 1st motion (Table 4-2a) most runs solved even when the initial guess error was 100%. Motions 2 and 3 (Tables 4-2b and 4-2c respectively) exhibit multiple solutions (4); all output errors not 0% or 100% represent solved runs where the computed solution differs from the correct solution. For example, using type 1 motion and a temporal extent of 0–0.9, we obtain two multiple solutions. The first occurs when an initial guess of 14.29% is used and is specified as:

U_l	$\vec{s}_l$	$\vec{\alpha}_l$	$\vec{\omega}_l$
-36.016876	-0.032824	0.294579	-0.088051
-85.635191	-0.078043	0.626386	0.145430
467.118290	0.425703	0.721709	0.022624

with X_{l3}=1090.651. The output error is 45.47%. This solution, plus the correct solution of $\vec{s}_l$=(0,0,0.4969759), $\vec{\alpha}_l$=(0,0,1) and $\vec{\omega}_l$=(0,0.1,0) produce the same 4 1/2 image velocities at the five image points and times:

y_1	y_2	time	v_1	v_2
70	70	0.0	0.02799	0.12799
-30	70	0.225	-0.17983	0.15677
-30	-30	0.45	-0.20499	-0.07606
70	-30	0.675	0.04999	-0.08645
20	20	0.9	-0.11230	-

(The last v_2 component for (20,20) is not used.) Except for these common image velocities, the two flow fields are distinct.

A third experiment that investigates the relationship between image velocity error and initial guess error does not produce any unexpected results; the two are usually independent. In most solved cases where the output error did not depend on image velocity error alone it was impossible to tell how much of the error was due to image velocity error and how much may have been due to the existence of a multiple solution.

As we have already seen in experiment 1, increasing temporal extent can reduce output error. The fourth experiment investigates what happens when spatial extent is varied from 7^o to 70^o (the full image) for a fixed temporal extent of 0–1. A spatial extent of θ^o is computed by first calculating $y=(\frac{1-\cos\theta}{2+2\cos\theta})^{1/2}$ and then using image velocities measured at (y,y) at time 0, $(-y,y)$ at time 0.25, $(-y,-y)$ at time 0.5, $(y,-y)$ at time 0.75 and (0,0) at time 1. The solution point for this experiment was changed to (0,0). The 1st motion is used and relative worst case image velocity error is varied from 0–1.4%. The results (Table 4.4a) show that increasing spatial extent increased output error in most cases. When $\vec{\alpha}_l$ was changed to (0.707107,0,0.707107) results (Table 4.4b) showed a slight improvement in output error for increasing spatial extent. Previous results with our monocular algorithm [Barron et al 87a] showed a better improvement with increasing spatial extent. We also investigate what happens when the spatial baseline $\vec{s}$ is varied: s_1 values of -1000, -800, -80, -8, 8, 80, 800 and 1000 are used. s_2 and s_3 remain 0; only horizontal disparity is used. Because ω_2 is 0.1, $-s_1$ values cause the right camera to move away from the planar surface (relative to the left camera) thus resulting in $\vec{U}_r > \vec{U}_l$ while $+s_1$ has the opposite effect, i.e. $\vec{U}_r < \vec{U}_l$. Relative worst case image velocity error is varied from 0–1.4% and a fixed temporal extent of 0-1 is used. Results (Table 4-4c) for the 3rd motion show that as $\vec{U}_r$'s value increases relative to $\vec{U}_l$ output error is reduced. Increasing the spatial baseline does help provided the increases cause the right camera to move faster and closer to the plane.

(4) [Subbarao and Waxman 85] show uniqueness of the monocular motion and structure parameters over time. Their result also holds when binocular flow fields are used. However, in both cases uniqueness holds only when the whole flow field is analyzed.

In the fifth experiment, we also wary the relative orientation of the left and right cameras by varying θ_2 to have values (in radians) of -0.5, -0.25, -0.1. -0.05,0.05,0.1,0.25 and 0.5. ϕ_1 and*, remain 0; the orientation involves a simple rotation about the X_2 axis. Again, relative worst case image velocity error of 0-1.4% and a fixed temporal extent of 0-1 are used. Results for the l* motion (Table 4.5) show that both convergent and divergent stereo setups yield smaller output error than for the original parallel setup; the closer the setup becomes to being parallel, the worst the output error.

Another result not included in this paper for lack of space shows that image velocity error caused by violation of the underlying assumptions produces less output error than similarly scaled random and worst case image velocity: it seems that violation of the various assumptions is less important than the accuracy of the input image velocities.

5 Conclusions

We have formulated a binocular reconstruction algorithm that uses a stereo spatio-temporal distribution of image velocities in left and right stereo image sequences but does not require point-to-point correspondence be solved in either the individual image sequences or between stereo image pairs. We have demonstrated that the addition of a temporal distribution of image velocity may increase the numerical stability of the solution technique. In addition, it allows us to analyze flow fields that may not be analyzable at one time. As well, increasing spatial extent can improve the algorithm's performance. Other results suggest that convergent/divergent stereo setups can give better results than parallel stereo setups and that increasing the spatial baseline can have a similar effect. In all cases, we are effectively increasing the spatio-temporal extent. Unfortunately, the greater the spatio-temporal extent the more likely the algorithm's underlying assumptions will be violated in realistic situations. We are able to report the existence of multiple solutions, a fact that is apparently overlooked by most other researchers. Our results indicate mat reconstruction techniques are quite sensitive to input image velocity error (1.4% maximum input error in the image velocities will be very difficult to obtain) but relatively insensitive to initial guess error. We believe that this is the main stumbling block that reconstruction algorithms have to overcome before we can consider this part of machine vision solved.

We are investigating the relationship between error in image velocities and error in the spatio-temporal derivatives of the flow fields [Barron et al 87b] and the improvement gained when a least squares formulation is used. These and other results will be reported in future papers.

Acknowledgements

We gratefully acknowledge financial support from the National Science and Engineering Research Council of Canada and the Department of Computer Science at the University of Toronto.

Bibliography

(1) Adiv G., 1984, "Determining 3-D Motion and Structure from Optictl Flow Generated by Several Moving Objects", COINS Technical Report 84-07, University of Massachusetts, April.
(2) Aloimonos J Y and I Rigoutsos, 1986, "Determining the 3-D Motion of a Rigid Planar Patch Without Correspondence, Under Perspective Projection, Proc. Workshop on Motion: Representation and Analysis, May 7-9.
(3) Arfken G., 1970, Mathematical Methods For Physicists, 2" Edition, Academic Press.
(4) Bandopadhay A. and J.Y. Aloimonos, 1985, "Perception of Rigid Motion from Spatio-Temporal Derivatives of Optical flow", TR 157, Dept. of Computer Science, University of Rochester, NY, March.
(5) Barnard ST. and M.A. Fischler 1982,, "Computational Stereo", ACM Computing Surveys, Vol. 14, No. 4, Dec., pp553-571.
(6) Barron, J., 1984, "A Survey of Approaches for Determining Optic Flow, Environmental Layout and Egomotion", RBCV-TR-84-5, Dept of Computer Science, University of Toronto, November.
(7) Barron, J., 1987, "Determination of Egomotion and Environmental Layout From Noisy Time-Varying Image Velocity in Monocular and Binocular Image Sequences", forthcoming PhD thesis, Dept of Computer Science, University of Toronto.
(8) Barron, J.L., A.D. Jepson and J.K. Tsotsos, 1987a, "Determining Egomotion and Environmental Layout From Noisy Time-Varying Image Velocity in Monocular Image sequences", submitted for publication.

(9) Barron, J.L., A.D. Jepson and J.K. Tsotsos, 1987a, "The Sensitivity of Motion and Structure Computations", Proc. of AAAI87, Seattle.

(10) Buxton B.F, H. Buxton, D.W. Murray and N.S. Williams, 1984, "3-D Solutions to the aperture Problem", in **Advances in Artificial Intelligence**, T. O'Shea (editor), Elsevier Science Publishers B.V. (North Holland), pp631-640.

(11) Jenkin M., 1984, "The Stereopsis of Time-Varying Images", RBCV-TR-84-3, Dept. of Computer Science, University of Toronto.

(12) Kanatani K, 1985, "Structure from Motion without Correspondence: General Principle", Proceedings of IJCAI, pp886-888.

(13) Lounguet-Higgins H.C. and K. Prazdny, 1980, "The Interpretation of a Moving Image", Proc. Royal Society of London, Vol. 208, Series B, pp385-397.

(14) Poggio G.F. and T. Poggio, 1984, "The Analysis of Steropsis", Annual Reviews of Neuroscience, Vol. 7, pp379-412.

(15) Snyder M.A., 1986, "The Accuracy of 3-D Parameters in Correspondence-Based Techniques: Startup and Updating", Proc. Workshop on Motion: Representation and Analysis, May 7-9.

(16) Subbarao M. and A.M. Waxman, 1985, "On the Uniqueness of Image Flow Solutions for Planar Surfaces in Motion", CAR-TR-114 (CS-TR-1485), Center for Automation Research, University of Maryland. (Also, 3rd Workshop on Computer Vision: Representation and Control, 1985.)

(17) Subbarao M., 1986a, "Interpretation of Image Motion Fields: A Spatio-Temporal Approach", Proc. Workshop on Motion: Representation and Analysis, May 7-9.

(18) Subbarao M., 1986b, "Interpretation of Visual Motion: A Computational Study", CAR-TR-221 (CS-TR-1706), Center for Automation Research, University of Maryland.

(19) Waxman A.M. and S. Ullman, 1985, "Surface Structure and 3-D Motion from Image Flow Kinematics", Intl. Journal of Robotics Research, Vol 4, No. 3, pp72-94.

(20) Waxman A.M. and K. Wohn, "Contour Evolution, Neighbourhood Deformation and Global Image Flow: Planar Surfaces in Motion", Intl Journal of Robotics Research, Vol.4, No.3, pp95-108.

(21) Waxman A.M. and K. Wohn, 1986, "Image Flow Theory: A Framework for 3-D Inference from Time-Varying Imagery", in **Advances in Computer Vision**, (ed C. Brown), Erlbaum Publishers, January.

(22) Waxman A.M. and J.H. Duncan, 1986, "Binocular Image Flows: Steps Toward Stereo-Motion Fusion", PAMI, Vol. 8, No. 6, Nov., pp715-729.

(23) Webb J.A. and J.K. Aggarwal, 1981, "Visually Interpreting the Motion of Objects in Space", IEEE Computer, August, pp40-46.

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6.14	4.05	3.55	5.62	0.93	2.14	0.56	2.07
16.64	5.20	7.88	4.66	8.44	0.08	2.58	2.44
13.67	28.21	7.35	4.01	3.59	3.92	6.63	3.79
48.46	26.26	9.94	11.08	16.05	8.24	1.19	0.18
15.34	46.82	13.68	11.05	3.07	3.67	7.69	3.54
29.88	3.38	17.68	20.42	17.45	3.54	6.66	0.92
2.08	74.09	96.08	9.83	65.72	26.43	8.75	10.61

Table 4.1a (1st motion Random)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24.16	17.12	10.73	7.48	6.52	4.47	3.78	3.24
77.33	37.47	29.06	19.22	13.63	10.13	8.48	6.30
100.00	73.77	39.86	31.97	22.91	15.11	11.83	9.61
100.00	100.00	68.42	45.92	33.52	21.47	16.99	13.17
100.00	100.00	109.69	70.69	50.06	32.83	24.81	16.53
100.00	100.00	100.00	110.90	58.65	40.66	27.50	23.33
100.00	100.00	100.00	100.00	100.32	57.71	42.67	26.78

Table 4.1b (1st motion Worst)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30.37	13.77	3.71	5.74	2.54	3.63	11.25	3.18
8.57	9.95	2.57	8.40	100.00	100.00	100.00	100.00
100.00	100.00	12.40	100.00	100.00	100.00	100.00	3.03
100.00	100.00	100.00	10.06	100.00	100.00	8.43	6.92
100.00	100.00	4.03	100.00	0.80	100.00	2.27	0.02
100.00	100.00	100.00	100.00	100.00	4.31	100.00	10.28
100.00	100.00	100.00	3.57	100.00	100.00	100.00	100.00

Table 4.1c (2nd motion Random)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21.70	16.40	14.07	100.00	100.00	100.00	6.02	4.66
100.00	100.00	100.00	100.00	100.00	100.00	100.00	6.72
100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table 4.1d (2nd motion Worst)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17.09	13.95	11.95	9.78	8.85	7.65	6.53	6.23
27.99	22.95	20.02	16.97	15.44	13.46	12.11	10.15
100.00	100.00	25.15	22.27	19.68	17.10	15.52	14.65
100.00	100.00	100.00	26.44	23.55	21.22	19.74	17.39
100.00	100.00	100.00	100.00	100.00	25.22	22.24	20.09
100.00	100.00	100.00	100.00	100.00	100.00	24.26	23.21
100.00	100.00	100.00	100.00	100.00	100.00	100.00	24.47

Table 4.1e (3rd motion Random)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17.09	13.95	11.95	9.78	8.85	7.65	6.53	6.23
27.99	22.95	20.02	16.97	15.44	13.46	12.11	10.15
100.00	100.00	25.15	22.27	19.68	17.10	15.52	14.65
100.00	100.00	100.00	26.44	23.55	21.22	19.74	17.39
100.00	100.00	100.00	100.00	100.00	25.22	22.24	20.09
100.00	100.00	100.00	100.00	100.00	100.00	24.26	23.21
100.00	100.00	100.00	100.00	100.00	100.00	100.00	24.47

Table 4.1f (3rd motion Worst)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.73	2.05	2.87	1.89	3.76	1.86	1.85	3.78
3.39	3.99	5.49	3.68	7.89	3.64	3.63	7.92
4.96	5.83	7.89	5.40	12.47	5.34	5.32	12.50
6.47	7.58	10.10	7.04	17.56	6.96	6.95	17.58
7.91	9.23	12.14	8.61	23.25	8.52	8.50	23.24
9.28	10.81	14.03	10.11	29.66	10.02	10.00	29.59
10.60	12.31	15.78	11.55	36.93	11.45	11.43	36.76

Table 4.4a (Varying Spatial Extent)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.01	2.09	3.63	4.18	1.39	1.11	0.88	0.69
3.90	4.05	6.86	8.97	2.73	2.18	1.74	1.36
5.69	5.91	9.75	14.49	4.02	3.23	2.58	2.02
7.39	7.66	12.34	20.92	5.26	4.24	3.40	2.67
9.01	9.32	14.69	28.52	6.46	5.23	4.20	3.30
10.53	10.90	16.82	37.64	7.62	6.18	4.98	3.93
11.99	12.39	18.76	48.76	8.74	7.11	5.75	4.55

Table 4.4b (Varying Spatial Extent)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100.00	100.00	100.00	0.00	100.00	0.00	100.00	100.00
100.00	100.00	0.00	100.00	100.00	0.00	0.00	100.00
100.00	0.00	100.00	0.00	100.00	100.00	100.00	100.00
0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00
100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table 4.2a (1st motion Initial Guess)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16.13	0.00	100.00	0.00	0.00	0.00	45.47	100.00
8.13	2.85	100.00	5.22	100.00	0.00	1.82	100.00
100.00	5.65	4.19	3.88	0.24	0.00	0.00	100.00
8.13	5.65	0.00	3.27	2.64	100.00	100.00	0.00
0.00	0.00	100.00	100.00	1.69	0.00	100.00	1.52
0.00	100.00	3.88	100.00	100.00	100.00	100.00	100.00
100.00	100.00	100.00	100.00	100.00	2.18	100.00	8.57

Table 4.2b (2nd motion Initial Guess)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00
100.00	100.00	100.00	0.00	100.00	0.00	0.00	100.00
0.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00
100.00	100.00	100.00	100.00	100.00	0.00	100.00	100.00
132.69	100.00	100.00	100.00	100.00	100.00	100.00	0.00
100.00	0.00	100.00	100.00	100.00	100.00	100.00	52.76

Table 4.2c (3rd motion Initial Guess)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.95	3.30	2.96	2.87	2.64	2.00	2.19	1.97
3.77	5.98	6.19	5.40	4.76	4.43	4.70	4.49
5.74	9.46	9.30	8.11	7.21	6.96	7.58	7.31
7.22	14.28	12.19	12.96	11.04	9.90	9.94	8.63
9.92	18.21	17.53	15.76	14.42	12.36	12.27	13.11
11.20	24.09	23.68	19.17	20.02	15.86	16.15	14.31
13.59	25.41	25.13	26.97	19.44	18.14	21.27	16.61

Table 4.4c (Varying s_1)

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.47	0.38	1.07	1.73	3.53	1.69	1.20	0.72
1.06	0.96	2.24	3.21	6.15	3.82	2.32	1.58
1.59	1.57	3.21	4.85	7.48	5.16	3.91	2.65
1.94	1.42	4.23	7.38	11.76	7.42	4.36	3.16
3.01	2.47	5.74	8.77	13.58	8.05	5.40	4.47
3.21	2.99	7.45	10.75	17.59	10.94	7.72	4.19
4.44	3.39	7.82	14.17	16.51	11.76	8.45	4.97

Table 4.5 (Varying ϕ_2)