Model Based Multiresolution Motion Analysis

James Beacom, Roland Wilson
Department of Computer Science,
University of Warwick,
Coventry,
CV4 7AL
email: jamesb@dcs.warwick.ac.uk
tel: 01203 522438
fax: 01203 525714

July 23, 1997

Abstract

A model based multiresolution motion analysis technique is presented. The projection onto the image plane of the motion of a point in three space is used in a derivative based optical flow formulation. This can then be solved using the method of least squares to compute the six parameters describing the motion of the body (three rotational and three translational components) initially at a high scale and then refined with increasing resolution These may be calculated due to prior knowledge of the depth of points in the scene from an explicit volumetric model of the object being viewed. This is specified at a single resolution and then extrapolated through scale. This enables the creation of a texture mapped approximation of the input frames at each scale to be used in the coarse-to-fine estimation of the motion parameters. The object is assumed not to deform through time.

Contents

List of Figures

1 Introduction

The analysis of motion has many applications. It is a key element in video compression, where it is used for prediction of subsequent frames and finds many other applications including robot guidance and object tracking.

The analysis starts from a pixel resolution motion field due to the change in grey level intensity in the image and is often referred to as optical flow [16]. It should be noted that this does not always correspond to the true projected scene motion due to a combination of the illumination characteristics and the lack of surface texture on certain objects. It is however what is observed in an image sequence.

A variety of techniques have been proposed for the analysis of motion. These can be broadly classified according to whether they attempt to estimate a dense flow field at all points in the images, which necessitates the use of additional constraints such as smoothness of variation in the flow field, or for a sparse set of features, which entails the problem of establishing correspondence between features in the two images [1] [10] [14] [15]. The correspondence problem was noted by Marr [11] and also exists in other areas of vision including stereo and image registration. It will not be considered further here. Dense flow fields generally rely on the use of additional constraints such as smoothness of variation in the flow field [6]. These are required because of the so called *aperture problem*, that only one component of the flow field may be measured directly, that in the direction of the luminance gradient at that point. This regularising term (the smoothness constraint) is balanced against the problem constraint (in this case the spatiotemporal derivative formulation to be explained in section 2) to overcome this ill-posedness.

Presented here is a new algorithm for motion estimation of rigid bodies within a multiresolution framework. Section 2 presents an overview of the least squares approach to three dimensional motion estimation based on a spatiotemporal derivative formulation of optical flow at a single scale. Section 3 explains the multiresolution volumetric model representation to be employed. Section 4 then explains the new algorithm which couples the model repsentation and the least squares motion estimation to propagate motion parameters in a coarse to fine manner through scale. Results are reported in section 5. Section 6 discusses the implications of the various approaches taken here, some issues to be resolved and expands on some future directions that this work may take.

2 Motion Estimation at a Single Resolution

2.1 The Geometry of Motion

As stated above, the present work currently relies on the assumption of rigid body motion, that is, the motion of a point from $\vec{x}$ to $\vec{x}'$ may be represented by

$$\vec{x}' = \mathbf{R}\vec{x} + \mathbf{T} \tag{1}$$

where $\mathbf{R}$ represents a rotation in terms of the Eulerian angles and is a product of three matrices, each corresponding to a rotation about one axis,

$$\mathbf{R} = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & 0 & -\sin\psi \\ 0 & 1 & 0 \\ \sin\psi & 0 & \cos\psi \end{pmatrix} \tag{2}$$

and $\mathbf{T}$ is the translational component of the motion $\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$. Under the assumption that the changes in Euler angles are small (denoted by $\Delta\phi$, $\Delta\theta$ and $\Delta\psi$) a linearisation can be introduced using the approximations $\sin\Delta\phi \approx \Delta\phi$, and $\cos\Delta\phi \approx 1$. Equation 2 then becomes

$$\begin{aligned}
\mathbf{R} &= \begin{pmatrix} 1 & \Delta\phi & -\Delta\psi \\ -\Delta\phi & 1 & \Delta\theta \\ \Delta\psi & -\Delta\theta & 1 \end{pmatrix} \\
&= \mathbf{I} + \begin{pmatrix} 0 & \omega_z & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{pmatrix} \Delta t \qquad (3)
\end{aligned}$$

where ω_x, ω_y and ω_z are the angular velocities about the axes and $\mathbf{I}$ is the identity matrix. This yields the rigid body motion equations (under the assumption of infinitesimal motion)

$$\begin{aligned}
x' &= x + \omega_z y\Delta t - \omega_y z\Delta t + v_x\Delta t \\
y' &= y - \omega_z x\Delta t + \omega_x z\Delta t + v_y\Delta t \\
z' &= z + \omega_y x\Delta t - \omega_x y\Delta t + v_z\Delta t \qquad (4)
\end{aligned}$$

A perspective projection is defined in terms of the focal length of the camera f, the image plane coordinates (X, Y) and the world coordinates (x, y, z) by similar triangles as

$$\begin{aligned}
X &= xf/z \\
Y &= yf/z \qquad (5)
\end{aligned}$$

The projection of a three dimensional motion onto the two dimensional image plane (u, v) under a perspective projection (assuming $f = 1$) for infinitesimal motion is given by [9] [13]

$$\begin{aligned}
\begin{pmatrix} u \\ v \end{pmatrix} &= \begin{pmatrix} \frac{dX}{dt} \\ \frac{dY}{dt} \end{pmatrix} \\
&= \begin{pmatrix} \frac{X' - X}{\Delta t} \\ \frac{Y' - Y}{\Delta t} \end{pmatrix} \\
&= \begin{pmatrix} XY & -(1+X^2) & Y & X/z & 0 & 1/z \\ (1+Y^2) & -XY & -X & -Y/z & 1/z & 0 \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \\ t_z \\ t_y \\ t_x \end{pmatrix} \\
&= \mathbf{CU} \qquad (6)
\end{aligned}$$

2.2 Least Squares Solution Using Optical Flow

Optical flow is the change in image intensity due to apparent motion [16]. Its computation is ill-posed in that only one component may be calculated, that in the direction of the luminance gradient at that point. This is commonly referred to as the *aperture problem.* Additional constraints must thus be used if both components are to be recovered. In this work, optical flow is not computed per se, but is used in conjunction with (6) above and a priori knowledge available from a model of the object being viewed. There exist many approaches to optical flow computation (e.g. [1] [2] [7] [10] [6] [5]) and the approach used here is derivative based (see [6]) which yields the brightness constraint equation

$$\mathbf{f}_x u + \mathbf{f}_y v + \mathbf{f}_t = 0 \qquad (7)$$

where $\mathbf{f}_x$, $\mathbf{f}_y$ and $\mathbf{f}_t$ denote the derivatives of the image vector $\mathbf{f}$ with respect to the spatial and temporal coordinates (i.e. $\frac{\partial\mathbf{f}}{\partial x}$, $\frac{\partial\mathbf{f}}{\partial y}$, and $\frac{\partial\mathbf{f}}{\partial t}$ respectively). This can be used in conjunction with (6)

motion parameter	true value	estimated value
ω_x	0.0	0.0007
ω_y	0.0	-0.0004
ω_z	0.0	0.0013
t_z	0.0	6.0845
t_y	-9.0	-1.0354
t_x	0.0	-0.2036

Table 1: Motion parameters estimated at the highest resolution only

to give

$$(\nabla \mathbf{f})^T \mathbf{C}\mathbf{U} + \mathbf{f}_t = 0 \tag{8}$$

where $\nabla \mathbf{f}$ is the vector of spatial derivatives. Let $\mathbf{G} = (\nabla \mathbf{f})^T \mathbf{C}$, then (8) can be written as the linear equation

$$\mathbf{G}\mathbf{U} = -\mathbf{f}_t \tag{9}$$

Defining a cost function

$$E = \| \mathbf{G}\mathbf{U} + \mathbf{f}_t \|^2 \tag{10}$$

the least squares solution for the motion parameters may be obtained as follows

$$\frac{\partial E}{\partial \mathbf{U}} = 2\mathbf{G}^{\mathbf{T}}(\mathbf{G}\mathbf{U} + \mathbf{f}_t) = 0 \tag{11}$$

Therefore

$$2\mathbf{G}^{\mathbf{T}}\mathbf{G}\mathbf{U} + 2\mathbf{G}^{\mathbf{T}}\mathbf{f}_t = 0 \tag{12}$$

$$\mathbf{U} = -(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{f}_t \tag{13}$$

This is the basis of the approach in [9]. The problem with this formulation is the inadequacy of the linearisation used to derive it. For example Table 1 shows the estimates from the application of (13) in the image plane i.e. at a single scale. The accuracy of the estimates can be improved by iteration, provided the initial error is not too large, but a more robust approach is to use a multiresolution method.

3 Model Representation

As stated above, motion may be computed as a consequence of a priori knowledge about the depth of points in the scene. Alternatively work is ongoing in trying to simultaneously estimate motion and the 3-d structure of the scene (e.g. [12]). In the current work depth of points is assumed known. This is due to the use of a model of the object under observation. The model is a volumetric representation in which voxels are assigned a numeric value indicating their level of membership of the object. This will enable the model to be extrapolated through scale thus allowing multiresolution motion estimation. The scale-space model is constructed by a process of successive smoothing and sub-sampling analogous to low-pass image pyramids (e.g. [3]) to construct an oct-tree.

So the value of a node in the tree (i.e. a voxel (i, j, k) at level l) where the base is regarded as level 0, and using a 5x5x5 Gaussian weighting function for the smoothing, is calculated from

$$f(i, j, k, l) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} \sum_{o=-2}^{2} w(m, n, o) f(2i + m, 2j + n, 2k + o, l - 1) \tag{14}$$

The smoothing mask is built from the separable filters suggested by Burt and Adelson [3]. In one dimension the filter coefficients are $(0.05, 0.25, 0.4, 0.25, 0.05)$.

In order to obtain an image as a projection of a volume defined in terms of voxel's level of membership of an object it is necessary to define a projection rule as, due to the smoothing, the values of voxels at coarse scales may take values anywhere in the range from $0 - 100\%$ of object membership. In fact, all that is required is a depth map and this is obtained from the depth of the voxel whose object membership exceeds a pre-defined threshold along the line of sight from the centre of projection through each image pixel. To obtain the projected image itself, the three- dimensional gradient is calculated for this voxel and the resultant image intensity set as the cosine of the angle between the line of sight and this gradient vector. The gradient is computed by convolving the volume with three 3-d filters in the style of the Sobel edge detectors (thus the $(1, 2, 1)$ local smoothing across the axis whose derivative component is being computed and $(1, 0, -1)$ differencing along the direction in which the derivative is being calculated, see [4]), one for each component of the gradient. Under a perspective projection this will inevitably necessitate interpolation in three dimensions in order to obtain the values of voxels at sub-voxel locations. The image intensities are only required however for generating a synthetic sequence and are not used during the motion estimation algorithm. It should be noted that the model is only defined once, at the finest resolution. In this case the object membership function (the values at each voxel) will be binary. As a consequence of the smoothing, the model will only have distinct surfaces at this highest resolution.

4 The algorithm

The least squares approach outlined above computes the rigid body motion parameters at a single scale. The multiresolution approach propagates motion parameters through scale according to Equation (15).

$$\hat{\mathbf{U}}_l = \mathbf{U}_{l|l+1} + \mathbf{A}_l(\mathbf{U}_l - \mathbf{U}_{l|l+1}) \tag{15}$$

where

$$\mathbf{U}_l = -(\mathbf{G}_l^T \mathbf{G}_l)^{-1} \mathbf{G}^T (\mathbf{f}_t)_l \tag{16}$$

is the motion estimate at a single scale l. $\mathbf{U}_{l|l+1}$ is the propagated parameter vector from the level above $(l+1)$ and the second term in equation (15) is the innovation term on the current level l. The scale-space propagation of the motion estimation permits this linearisation of an essentially non-linear problem. In the simplest case, the Kalman gain, $\mathbf{A}_l$ is set to unity, but in general should take into account the effects of noise which will primarily affect the derivatives at the highest resolutions.

Given two frames of an image sequence, and a model of the object being viewed, the various pyramids are constructed; the oct-tree model representation described above (Section 3) and a low pass image pyramid constructed on each of the original images. Motion estimation is then computed at the coarsest resolution by fitting the motion parameters using least squares to the optical flow field obtained from the spatiotemporal derivatives (figure 1). These parameters are then used to reposition the model on the level below and create an estimate for the level below (figure 2). In fact, the model is first translated to the centre of the volume, before being rotated and translated to its final position. This avoids clipping which can otherwise result during the rotation. The estimate is created by texture mapping the appropriate level of the pyramid from the first input image using the new depth map (from the newly positioned model). This texture mapping is implemented according to a rearrangement of equation 6, Thus,

$$\begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} XY & -(1+X^2) & Y & X/z & 0 & 1/z \\ (1+Y^2) & -XY & -X & -Y/z & 1/z & 0 \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \\ v_z \\ v_y \\ v_x \end{pmatrix} \tag{17}$$

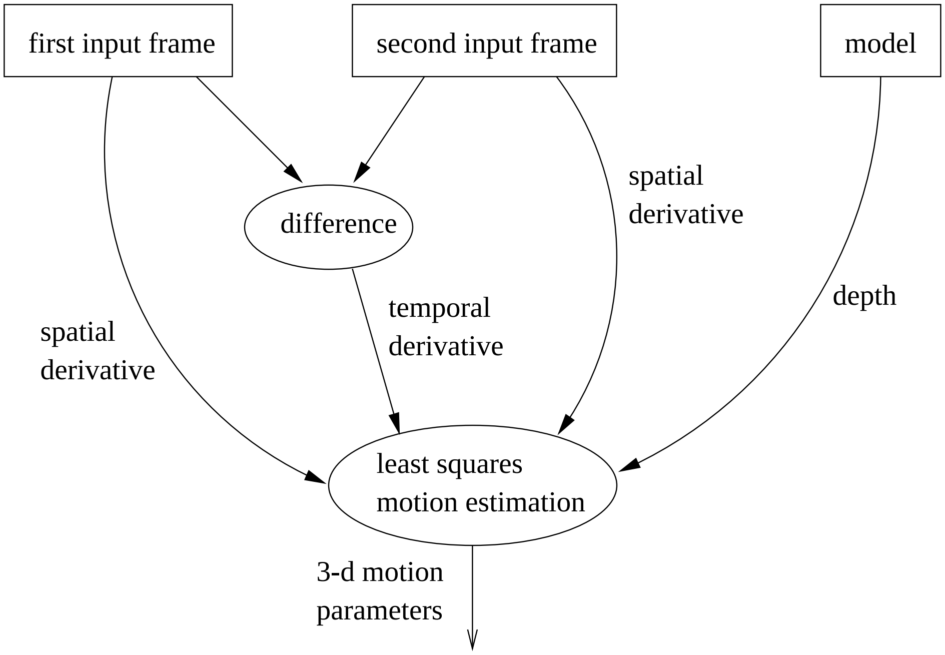

Figure 1: The initial motion estimation at the coarsest scale

where z is the *new* depth. This will calculate, for each pixel in the image, the location in the previous image (to sub pixel accuracy) from which it came. Its grey level may then be set using bilinear interpolation. Having created the estimate for the level below, the algorithm continues in a coarse to fine manner computing the motion parameters between the texture mapped images and the second frame low pass pyramid. These are then used to texture map the level below (Figure 2). The multiresolution approach results in only small values for the parameters being computed at each scale as on each level an approximation of the second frame is available (from the motion parameters of the level above).

Once the texture mapped image for the finest resolution has been created (using the parameters from the previous pyramid level) motion is computed again for the final time and this is used to create the texture mapped estimate of the original sequence. In summary,

1. Construct 2-d image pyramids on the two input frames.
2. Construct the pyramid for the model (in the correct initial position).
3. Compute the 3-d motion parameters using Equation 13 between the coarsest scales of the pyramids built on the input frames.
4. For l = coarsest resolution to 1,
   (a) Using the computed 3-d motion parameters from level $l+1$, rotate and translate the model on the current level l.
   (b) Use the depth from the newly positioned model, and the 3-d motion parameters from level $l+1$ to compute the 2-d projected motion field at level l.
   (c) Texture map level l of the 2-d pyramid built on the first input frame using the flow vectors computed above.

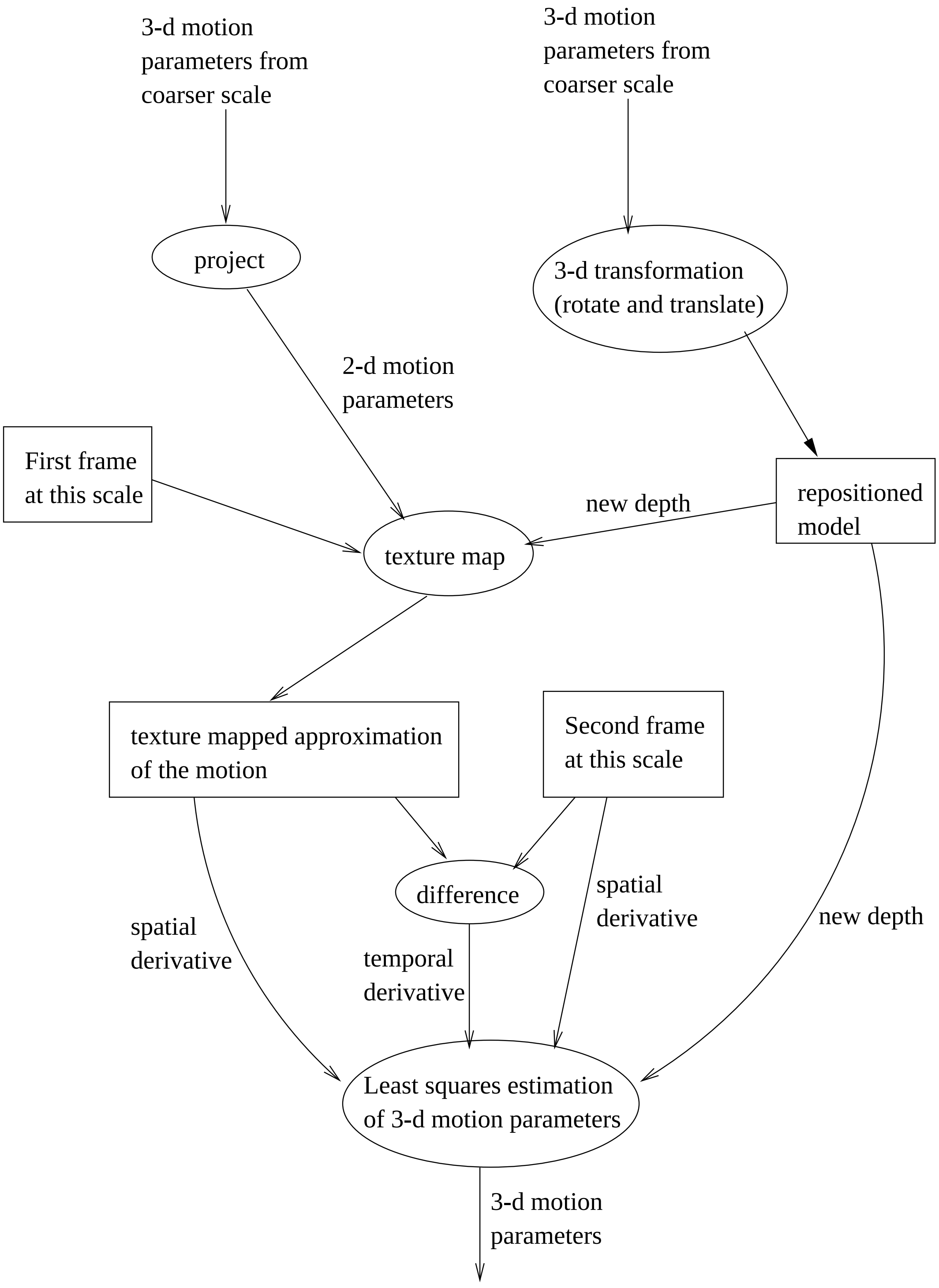
3-d motion parameters from coarser scale
3-d motion parameters from coarser scale
project
3-d transformation (rotate and translate)
2-d motion parameters
First frame at this scale
repositioned model
new depth
texture map
texture mapped approximation of the motion
Second frame at this scale
difference
spatial derivative
spatial derivative
new depth
temporal derivative
Least squares estimation of 3-d motion parameters
3-d motion parameters

Figure 2: Motion estimation at single scales (except the coarsest)

para-	true	level			
meter	value	3	2	1	0
ω_x	0.0	0.0	-0.0034	-0.0034	-0.0026
ω_y	0.0	-0.0032	0.0019	0.0024	0.0021
ω_z	0.15	0.1579	0.1464	0.1489	0.1507
t_z	0.0	-0.3315	-0.5292	-0.5834	-0.6057
t_y	0.0	-0.0680	1.2201	1.2875	0.9807
t_x	0.0	0.7836	1.5283	1.1795	0.9233

Table 2: the estimated motion and correct values at each scale relating to the rotation example

para-	true	level			
meter	value	3	2	1	0
ω_x	0.0	0.0065	0.0095	0.0123	0.0130
ω_y	0.0	-0.0045	-0.0045	-0.0048	-0.0051
ω_z	0.0	0.0421	0.0224	0.0138	0.0109
t_z	0.0	3.2318	3.3864	3.5470	3.5602
t_y	-9.0	-6.6451	-9.6957	-11.554	-11.729
t_x	0.0	-1.9098	-1.6109	-1.5608	-1.4522

Table 3: the estimated motion and correct values at each scale relating to the vertical translation example

(d) Compute the 3-d motion parameters using Equation 13 between the texture mapped image and level l of the 2-d pyramid built on the second input frame.

5. With the 3-d motion parameters, texture map the finest resolution of the first input image pyramid to create the final motion estimate.

A requirement of the algorithm is that the model is initially placed at the correct position of the object in the first frame.

5 Results

The results of this algorithm applied to synthetic sequences are shown in Figures 3 and 4. Displayed are the two input frames, the motion estimate (texture mapped), the error, and the flow needles indicating the locations from which each pixel originated (used in the texture mapping). Tables 2 and 3 show the motion parameters computed at each scale.

6 Discussion

Optical flow computed using the spatiotemporal derivatives of an image suffers from some well known problems [2]. These include the problem of how to compute the derivatives and aliasing inaccuracies with motions of more than half a pixel per frame. The multiresolution approach allows far greater displacements because at high scales the equivalent displacement is sufficiently small. Derivative calculation is approximated using adjacent differencing of pixels in the first image (for the spatial derivatives) using two 3×3 convolution kernels presented in [19] or between frames (for the temporal derivative). These methods are obviously susceptible to noise, especially the temporal derivative which involves no averaging (simply the difference between the pixel at the appropriate

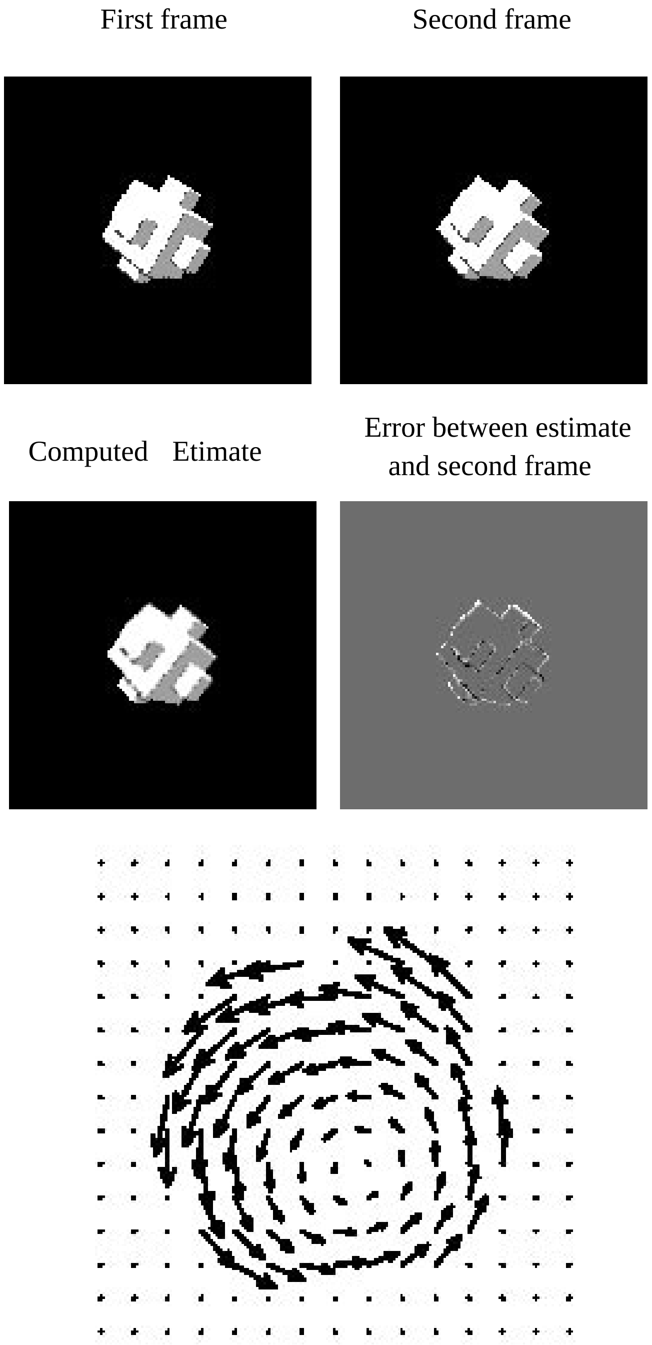

Figure 3: The output from the motion estimation algorithm for a rotation about the z-axis

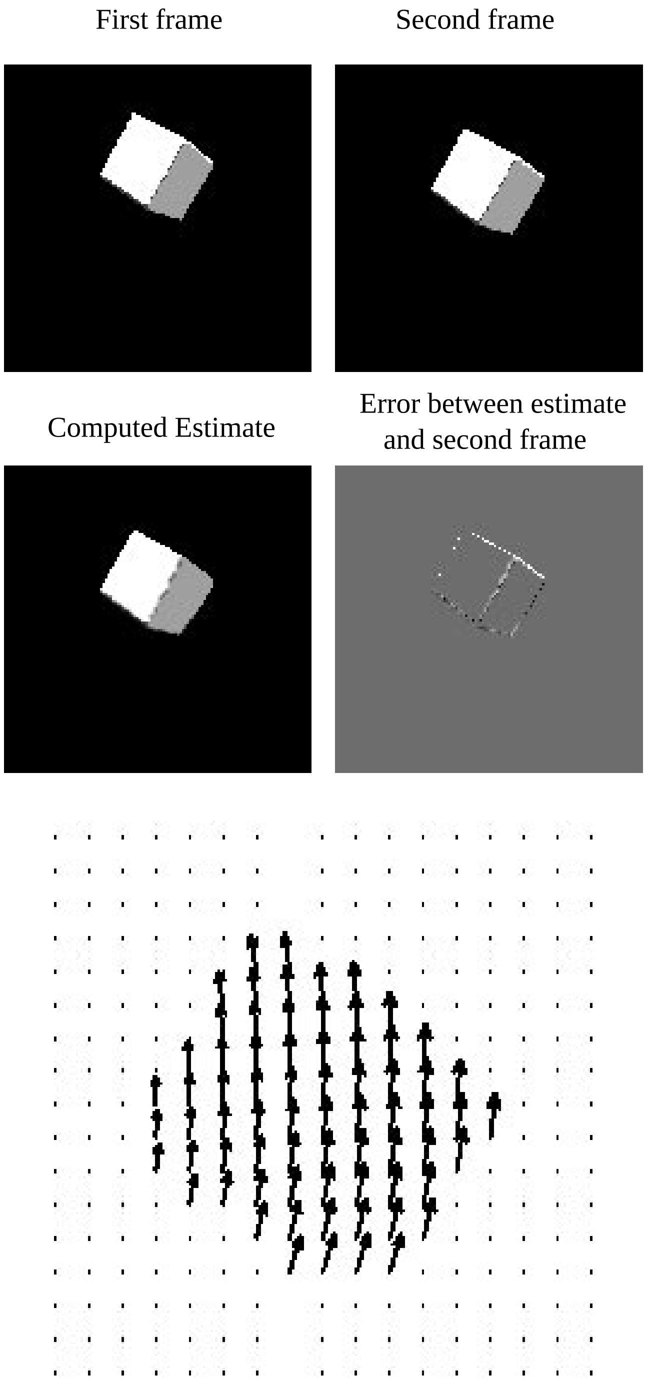

Figure 4: The output from the motion estimation algorithm for a vertical translation of the object

position in each frame). It would therefore be desirable to replace the derivative based approach with an alternative such as a correlation based approach (e.g. [8]).

Under the present formulation, a linearisation is introduced into the rotation matrix using the approximations $\sin\theta \approx \theta$ and $\cos\theta \approx 1$ for small θ. This allows the formulation of the linear Equation 9. However, this obviously results in increasing errors for larger rotations. An alternative might be to use the full rotation matrix without the simplifying approximations.

There are also issues to be addressed regarding the representation of the model. The current method relies on the use of a threshold to determine the exact location of the object boundaries at coarse scales as the object membership values of each voxel are only binary for the definition of the model at the highest resolution. Although it might be possible to use some form of three dimensional texture, it is unclear how this could be achieved. The most likely candidate to replace this would seem to be some form of mesh or deformable surface which could deform to the surface contours (e.g. [17]).

The initialisation of the model is yet to be resolved. Currently it is placed in the correct position (aligned with the object in the first frame). It may be possible to start with the model's pose known and calculate the motion between this and the first frame in the sequence.

Alternatively it may be feasible to simultaneously estimate the depth and motion parameters. [18] presents an iterative two stage algorithm in which the motion parameters are estimated at the first iteration and then used to update the depth estimation as the second step.

7 Conclusion

A new approach to motion estimation based on a multiresolution model has been presented. The model provides a priori knowledge of the scene depth thus enabling the calculation of three dimensional motion parameters without additional constraints. Results from synthetic image sequences have been presented and possible future directions for the work have been discussed.

References

[1] J. K. Aggarwal and N. Nandhakumar. On the Computation of Motion from Sequences of Images - A Review. *Proceedings of the IEEE*, 76:917–935, 1988.

[2] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12:43–77, 1994.

[3] P. J. Burt and E. H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications*, 31:337–345, 1983.

[4] R. G. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[5] D. J. Heeger. Optical Flow Using Spatiotemporal Filters. *International Journal of Computer Vision*, pages 279–302, 1988.

[6] B. K. P. Horn and B. G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.

[7] J. K. Kearney, W. B. Thompson, and D. L. Boley. Optical Flow Estimation: An Error Anaylsis of Gradient-Based Methods with Local Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:229–244, 1987.

[8] S. A. Kruger and A. D. Calway. A Multiresolution Frequency Domain Method for Estimating Affine Motion Parameters. In *IEEE International Conference on Image Processing*, pages 113–116, 1996.

[9] H. Li, P. Roivainen, and R. Forchheimer. 3-D Motion Estimation in Model-Based Facial Image Coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:545–555, 1993.

[10] M. R. Luettgen, W. Clem Karl, and A. S. Willsky. Efficient Multiscale Regularization with Applications to the Computation of Optical Flow. *IEEE Transactions on Image Processing*, 3:41–64, 1994.

[11] D. Marr. *Vision, a computational investigation into the human representation and processing of visual information*. W. H. Freeman, 1982.

[12] P. F. McLauchlin and D. W. Murray. A Unifying Framework for Structure and Motion Recovery from Image Sequences. Technical report, Active Vision Lab, University of Oxford, Oxford, UK, 1995.

[13] A. N. Netravali and J. Salz. Algorithms for Estimation of Three-Dimensional Motion. *AT&T Technical Journal*, 64:335–346, 1985.

[14] G. L. Scott and H. C. Longuet-Higgins. An Algorithm for Associating the Features of Two Images. *Proceedings of the Royal Society of London*, 244:21–26, 1991.

[15] L. S. Shapiro. *Affine Analysis of Image Sequences*. Cambridge University Press, 1995.

[16] A. Singh. *Optic Flow Computation: A Unified Perspective*. IEEE Computer Society Press, 1991.

[17] A. J. Stoddart, A. Hilton, and J. Illingworth. Slime: A new deformable surface. In *British Machine Vision Conference*, pages 285–294, 1994.

[18] A. M. Tekalp. *Digital Video Processing*. Prentice Hall, 1995.

[19] R. Wilson and A. H. Bhalerao. Kernel Design for Efficient Multiresolution Edge Detection and Orientation Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:384–390, 1992.