Model-based Brightness Constraints: on Direct Estimation of Structure and Motion*

G. P. Stein

Artificial Intelligence Laboratory
MIT
Cambridge, MA 02139
gideon@ai.mit.edu

A. Shashua

Institute of Computer Science
Hebrew University of Jerusalem
Jerusalem 91904, Israel
http://www.cs.huji.ac.il/∼ shashua/

Abstract

We describe a new direct method for estimating structure and motion from image intensities of multiple views. We extend the direct methods of Horn & Weldon [20] to three views. Adding the third view enables us to solve for motion, and compute a dense depth map of the scene, directly from image spatio-temporal derivatives in a linear manner without first having to find point correspondences or compute optic flow.

We describe the advantages and limitations of this method which are then verified with experiments using real images.

1. Introduction

The geometry of multiple views of a 3D scene is well understood. There exist geometric constraints, which relate corresponding points and lines in multiple views to the camera geometry. These constraints take the form of the trilinear tensor equations for three views, and the epipolar constraints for two views. Given a set of feature correspondences we can recover the camera geometry, and in particular camera motion (or the relative position of multiple cameras). But finding correspondences is a hard problem, and the features must be recovered accurately in order to correctly recover camera motion.

Feature correspondence, whether optic flow or discrete features, is based in some form or another, on the constant brightness assumption. That is, we assume the same feature in two views share some photometric properties. In it's strictest form, the *constant brightness constraint*, assumes that the brightness of the corresponding point does not change between views. Alternatively, we might look at some function of the brightness, possibly a nonlinear function such as brightness edges. The constant brightness constraint is a good approximation for many surfaces in the real world, especially if the motion is small. However, the constant brightness constraint is not strong enough to give us true correspondences. Local measurements, for example, cannot provide 'optic flow' but only *normal flow*, that is, the image flow estimates in the direction of the image brightness gradient. This is a particular problem in scenes with long, nearly straight, edges. We will show some examples later in the paper.

In this paper we present the *model-based brightness constraints*, where we combine geometric motion models with the brightness constraint. This provides us with a stronger constraint that can be used for direct estimation of structure and motion. We can thereby bypass the correspondence problem, and recover the camera motion directly from the image brightness values. This results in accurate motion estimates and good 3D reconstruction in very challenging scenes.

We now provide the key ideas behind the new method. The *optic flow constraint equation* [19] provides a matching constraint between a point in one image, and a line passing through the corresponding point in the second image. In other words, for every point in one image it gives us the equation of a line along which the corresponding point must lie in the second image. By *point* we refer to every pixel in the image which has a none zero spatial derivatives (image gradient). The equation is given in terms of the spatio-temporal derivatives of the image brightness, and the line is parallel to the iso-brightness contour in the first image. We note that the 'optic flow constraint equation' is a first order approximation and assumes that the image motion (and hence typically, the camera motion) is small.

There are no geometric constraints on point-line correspondences between two views. This can be seen using the following reasoning. A point in the image defines a line in space. A line in the second image defines a plane in space. A line and a plane always intersect. Therefore, given a set of points in one image and a set of corresponding lines in the second image, for every camera geometry there exists a valid 3D interpretation and no constraint on the geometry exists. We must therefore use three view geometry, where a point in one view, and lines through the corresponding points in two other views, provide a constraint, which can be written in the form of the *trilinear tensor* [34, 15, 42]. The 27 coefficients of the 'trilinear tensor' encapsulate the camera motions and the internal parameters of the camera such as the focal length. The new method can therefore be viewed as a necessary extension of the 'direct methods' of Horn & Weldon [20] from two views (one motion) to three views (two motions). These methods are dubbed 'direct methods' because they do not require prior computation of optic flow.

By combining the optic flow constraint equation with the geometric model of the 'trilinear tensor' we obtain the

*We acknowledge US-IS BSF contract 94-00120 and the European ACTS project AC074. General support for G.S. comes from DARPA contracts N00014-94-01-0994 and 95009-5381.

tensor brightness constraint [45] that describes the relationship between the spatio-temporal brightness derivatives at each pixel in the image, with the camera parameters modeled by the 27 coefficients of the trilinear tensor. This 'tensor brightness constraint' provides one linear equation per pixel in the image which results in a highly over-constrained set of equations. The 'tensor brightness constraint' is valid for the most general case (projective) where the cameras undergo general motion and we do not know the internal camera parameters which might vary from frame to frame. We then proceed through a hierarchy of reduced motion models first by assuming calibrated cameras and then by assuming the Longuett-Higgins & Prazdny small motion model [23] resulting in reduced model-based brightness constraints for those motion models. Parts of this work, as it evolved, have been presented in technical reports and meetings found in [39, 45].

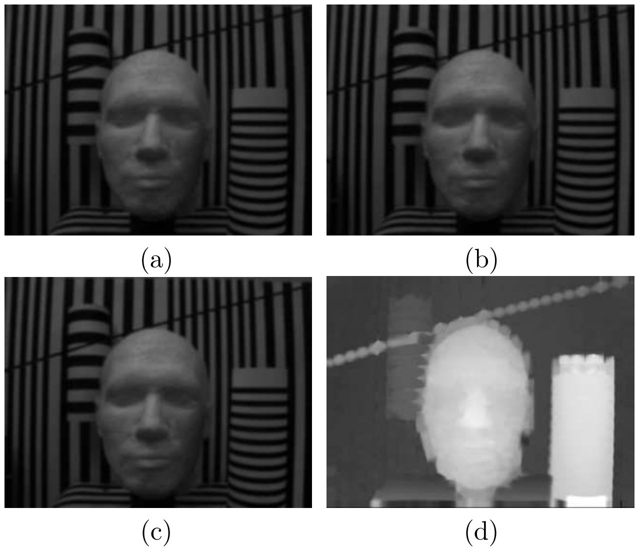


Fig. 1. *The three input images (a,b,c) and the estimated depth map (d). The motion between images (a) and (b) is horizontal. The motion is small but can be seen in the width of the rightmost vertical stripe and in the parallax between the cylinder on the right and the vertical stripes behind it. The motion of between images (a) and (c) was vertical.*

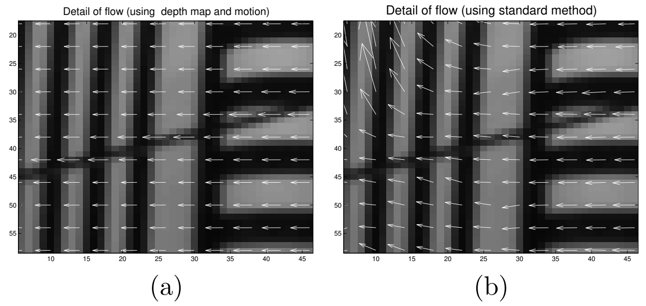


Fig. 2. *Detail of optic flow computed for Figure 1. (a) Correct optic flow computed using motion and depth map recovered by the direct method. The camera motion was horizontal parallel to the image plane. The flow vectors have a zero Y component. (b) Optic flow computed using Bergen & Hingorani optic flow program. Note in the upper left corner the flow vectors have a strong Y component. This is due to the aperture problem where even a large window (aperture) will see image gradients in only one direction giving no constraint on the Y component of the flow. Near the intersections (in the image) of the diagonal line and the vertical bars the flow has a small Y component. This error occurs because the program 'tracks' the intersection point as if it were a real 'feature' point, but the lines do not intersect in space. Although smaller in magnitude this is the more significant error because the program will also give a high confidence to this value.*

1.1. Related Work

1.1.1. Correspondence Based Methods

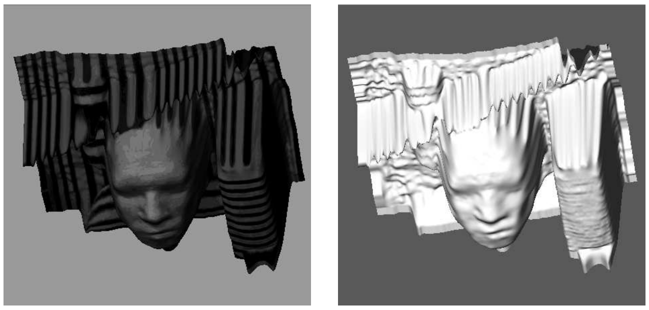
The standard approach to the problem of structure from motion is to first compute correspondences. These might be either dense correspondence in the form of optic flow [19, 25, 29] or feature correspondence [7, 51, 9]. Then the correspondences are used to compute the camera motion and scene structure. The advantage of our method over both optic flow methods [24, 25, 16, 32, 48] and feature based methods [49, 9, 40, 41, 2] is that no prior computation of correspondences is needed, a computation which in itself is error prone. Since we obtain a linear system of equations that combines the information from all the pixels in the image, we avoid the aperture problem without having to apply a smoothness assumption. By avoiding the need to explicitly detect feature points we can use information from areas where gradient information is weak such as shadow edges.

These advantages are highlighted in a scene such as in Figure 1. The scene contains a background of straight bars, a plaster bust in the foreground, and an oblique line cutting the bars. The background bars provide an unreliable source of information for point-to-point correspondence because of the aperture problem. Even large regions of the image have no information which constrains the flow in the Y direction. Many feature points in the image do not correspond to real feature points in the 3D world. For example, the intersection of the vertical bars and the diagonal line in the image do not correspond to physical features in 3D since the line does not lie on the background plane. Figure 2 shows a detail of the optic flow field in the upper left of the scene where these two problems occur. The flow was computed using a widely available, 'industrial strength', optic flow program [5, 4]. (For more examples see Section (5.3).)

Many natural scenes such as tree branches or man made objects such as window frames, lamp posts and fences, often give rise to these problems. Finally, the texture on the plaster bust varies smoothly and with low contrast thereby giving rise only to a relatively small number of reliable features to track. Yet, our method produces a full 3D model of the scene as demonstrated in Figure 3a,b.

1.1.2. Direct Methods

The 'direct methods' were pioneered by Horn & Weldon in [20]. Using only a single image pair they ended up with N equations in $N+5$ unknowns, where N is the number of points in the image. The unknowns are the N depth variables, the 3 translation and 3 rotation parameters with one unknown dropping out because translation and depth can only be found up to a common scale factor.



(a) (b)

Fig. 3. *3D rendering of the estimated depth map from images in Figure 1. These images show the inverse depth, $k = \frac{1}{z}$, which is the natural value to compute. In (b) the texture map was removed to show the detail and the flaws.*

The problem is therefore ill-posed and additional constraints are needed. Negahdaripour & Horn [30] present a closed form solution assuming a planar surface. Szeliski & Kang [47] describe an iterative solution using splines to enforce a smoothness constraint on the depth. McQuirk [26] shows that in a pure translation model the subset of the image points with a nonzero spatial derivative but a zero time derivative gives the direction of motion, thus the *focus of expansion* (FOE) is on a line perpendicular to the gradient at these points. Since this method uses only a subset of the image points, it ignores most of the image data.

Heel [17] constructs a Kalman filter to build up a structure model from more than one image pair but the core computation is fundamentally the same single image pair computation. The basic idea is that the Horn & Weldon equation is linear in depth if motion is known and linear in motion if depth is known. Heel initially assumes a depth (constant depth surface) and estimates the motion and then uses the motion to estimate the depth. The depth map is then warped using the computed motion and used as the initial depth estimate for the next image pair in the sequence. The question of convergence is not addressed and results are shown only for pure translation. Heel limits the motion to image motions of less than 3 pixels which is one of the limiting factors in the accuracy of the surface reconstructions. Within a coarse-to-fine implementation (Section 4.4) our method can handle much larger image motions averaging up to 50 pixels for 640×480 resolution images, thereby increasing the dynamic range by an order of magnitude..

Michaels [27] uses three frames (two motions). Each motion gives N equations totaling $2N$ equations. The unknowns are the N unknown depths and the 2×6 translation and rotation parameters. Again, one parameter drops out because of the translation and depth scale ambiguity. He solves the $2N$ equations with $N + 11$ unknowns as a large nonlinear optimization problem using the Levenberg-Marquart algorithm [28]. Since it took a very long time to converge results are shown for very low resolution images only. He also shows theoretically that a large field of view ($\approx 120^o$) is required for accurate direct estimation of motion given structure. Kumar et al [22] first align a dominant plane in the images. The residual image motion is epipolar motion (i.e. pure translation). The epipole of the residual motion is found using iterative techniques starting with an initial guess. Others who have explored planar alignment include [21, 33]. The theoretical framework for planar alignment can also be found in [37].

More recently, Fermuller & Aloimonos [11, 12] describe global geometric properties of the flow-field that give rise to direct relationships between the measurement of normal flow and the ego-motion parameters solved by means of search techniques. They apply the *depth is positive* constraint and try to find an optimally smooth surface.

2. Mathematical Background

2.1. Notation

We will use uppercase bold to denote matrices (e.g. **A**). Vectors representing 3D points which will be upper case and not bold. Other vectors and scalars will be in lower case.

We will occasionally use tensorial notations. We use the covariant-contravariant summation convention: a point is an object whose n coordinates are specified with superscripts, i.e., $p^i = (p^1, p^2, ..., p^n)$. These are called contravariant vectors. An element in the dual space (representing hyperplanes — lines in $\mathcal{P}^2$), is called a covariant vector and is represented by subscripts, i.e., $s_j = (s_1, s_2, ..., s_n)$. Indices repeated in covariant and contravariant forms are summed over, i.e., $p^i s_i = p^1 s_1 + p^2 s_2 + ... + p^n s_n$. This is known as a contraction. An outer-product of two 1-valence tensors (vectors), $a_i b^j$, is a 2-valence tensor (matrix) c_i^j whose i, j entries are $a_i b^j$ — note that in matrix form $C = ba^\top$.

Matching image points across three views will be denoted by p, p', p''; the homogeneous coordinates will be referred to as p^i, p'^j, p''^k, or alternatively as non-homogeneous image coordinates $(x, y), (x', y'), (x'', y'')$ — hence, $p^i = (x, y, 1)$, etc.

We will now consider three perspective views ψ, ψ' and ψ'' of a 3D scene. Figure 4 shows a 3D point $P \in \mathcal{P}^3$ and its image in the three views $p \in \psi$, $p' \in \psi'$ and $p'' \in \psi''$. Without loss of generality we can align the 3D world coordinate system with the coordinate system of the first camera:

$$p \;\cong\; \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \;\cong\; [\mathbf{I}; 0]\, P \tag{1}$$

Thus $P \cong (x, y, 1, \rho)^\top$. After we have set the first camera coordinate system, the other two camera coordinate

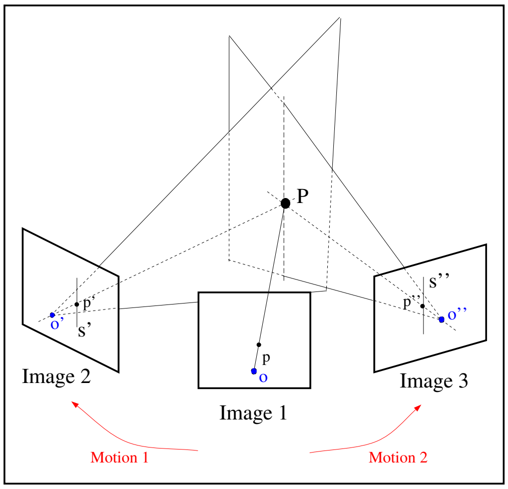


Fig. 4. *The Geometry of Three Views: Image 1, Image 2 and Image 3 are three views obtained from cameras centered at o, o' and o'' respectively. Point P in the scene projects to image points p, p' and p''. s' and s'' are any image lines passing through the points p' and p'' respectively.*

systems are in general given by:

$$p' \;\cong\; \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \;\cong\; \tilde{\mathbf{A}}P \;\cong\; [\mathbf{A}; t']\,P \;\cong\; \mathbf{A}p + \rho t' \tag{2}$$

$$p'' \;\cong\; \begin{pmatrix} x'' \\ y'' \\ 1 \end{pmatrix} \;\cong\; \tilde{\mathbf{B}}P \;\cong\; [\mathbf{B}; t'']\,P \;\cong\; \mathbf{B}p + \rho t'' \tag{3}$$

The matrices $\mathbf{A}$ and $\mathbf{B}$ are homography matrices from Image 1 to Image 2 and to Image 3 respectively *due to the same plane* π. The vectors t' and t'' are the epipoles, the projection center of camera 1 projected onto the image planes of the 2nd and 3rd cameras respectively. ρ is the relative affine depth[37]. It is important to note that ρ is independent of the second view point and thus has the same value in equations (2) and (3). In a calibrated setting, (i.e. if the intrinsic parameters are known), the matrices $\mathbf{A}$ and $\mathbf{B}$ are rotations and the vectors t' and t'' are the translations. Finally, in the calibrated setting ρ is replaced by $\rho = \frac{1}{z}$.

We will now derive the trilinear tensor of [34, 15, 42]. Let s' and s'' be lines through points p' and p'' respectively:

$$s'^{\top}p' = 0 \qquad\qquad s''^{\top}p'' = 0 \tag{4}$$

Premultiplying the left and right hand sides of (2) by $s'^{\top}$ and of (3) by $s''^{\top}$ we get:

$$\begin{array}{rcl} s'^{\top}\mathbf{A}p + \rho s'^{\top}t' & = & 0 \\ s''^{\top}\mathbf{B}p + \rho s''^{\top}t'' & = & 0 \end{array} \tag{5}$$

Eliminating ρ from the above equations results in the equation:

$$s'^{\top}t's''^{\top}\mathbf{B}p - s''^{\top}t''s'^{\top}\mathbf{A}p = 0 \tag{6}$$

This can be written compactly using tensor notation:

$$p^i s''_k s'_j \mathcal{T}_i^{jk} = 0 \tag{7}$$

where $\mathcal{T}$ is the tensor representing a bilinear function of the camera matrices:

$$\mathcal{T}_i^{jk} = t'^j b_i^k - t''^k a_i^j \tag{8}$$

Equation (7) relates a point p in Image 1 and lines s' and s'' passing through the corresponding points p' and p'' in Image 2 and Image 3 respectively. It is important to note that the lines s' and s'' do not have to correspond to any physical line in space or the image. More details can be found in a semi-review paper [35] and study on various aspects and applications can be found in [40, 38, 10, 50, 18, 36, 46, 1].

3. Model-based Brightness Constraints

3.1. Photometric Constraints

Geometrically, a trilinear matching constraint is produced by contracting the tensor with the point p in Image 1, *any* line coincident with p' in Image 2, and *any* line coincident with p'' in Image 3. In particular, we may use the tangent to the iso-brightness contour at p' and p'', respectively, and thus one can recover in principle the camera matrices across three views in the context of the "aperture" problem, as suggested by [42]. However, there still remains the problem of finding those matching tangents in the first place. This we now solve.

Assume the brightness constancy assumption,

$$I_2(x, y) = I_(x - u', y - v'),$$

where $I(x, y), I_2(x, y)$ are the observed grey-scale images at two successive time frames. Since the displacement u', v' are assumed to be small (infinitesimal motion assumption), then the equation above can be simplified through the truncated (first-order) Taylor series expansion of $I(x, y)$ to what is known as the "optic flow constraint equation" [19]:

$$u'I_x + v'I_y + I'_t = 0 \tag{9}$$

where I_x, I_y are the x, y spatial derivatives, respectively, and $I'_t = I_2(x, y) - I(x, y)$ is the temporal image derivative.

The optic flow constraint equation (9) can be rewritten in the form:

$$(I_x, I_y, I'_t)^{\top}(u, v, 1) = 0.$$

The line

$$s = (I_x, I_y, -xI_x - yI_y)^{\top}$$

in the projective plane passes through the point $p = (x, y, 1)^{\top}$ since:

$$(I_x, I_y, -xI_x - yI_y)^{\top}(x, y, 1) = 0.$$

Combining those two equations together:

$$\begin{array}{rcl} \left(\begin{array}{c} I_x \\ I_y \\ -xI_x - yI_y \end{array}\right)^\top \left(\begin{array}{c} x \\ y \\ 1 \end{array}\right) + \left(\begin{array}{c} I_x \\ I_y \\ I'_t \end{array}\right)^\top \left(\begin{array}{c} u \\ v \\ 1 \end{array}\right) & = & \\ \left(\begin{array}{c} I_x \\ I_y \\ I'_t - xI_x - yI_y \end{array}\right)^\top \left(\begin{array}{c} x' \\ y' \\ 1 \end{array}\right) & = & 0 \end{array} \quad (10)$$

Where we have used $x' = x + u$ and $y' = y + v$. Therefore, the line:

$$s' = \left(\begin{array}{c} I_x \\ I_y \\ I'_t - xI_x - yI_y \end{array}\right)^\top \quad (11)$$

passes through the point $p' = (x', y', 1)^\top$.

Thus the photometric constraints provide a matching constraint between a point p and a line s' passing through the corresponding point p' in Image 2, and between a point p and a line s'' passing through the corresponding point p'' in Image 3. In other words, the lines:

$$s' = \left(\begin{array}{c} I_x \\ I_y \\ I'_t - xI_x - yI_y \end{array}\right) \quad (12)$$

and

$$s'' = \left(\begin{array}{c} I_x \\ I_y \\ I''_t - xI_x - yI_y \end{array}\right) \quad (13)$$

are lines coincident with p' and p'' respectively and parallel to the iso brightness contour at (x, y). I''_t is the temporal derivative between the Image 3 and Image 1. (i.e. $I''_t = I_3(x, y) - I_1(x, y)$.)

3.2. The Projective Model: The Tensor Brightness Constraint

Substituting equations (12) and (13) into the tensor equation (8) results in the *tensor brightness constraint*:

$$\boxed{s''_k s'_j p^i \mathcal{T}_i^{jk} = 0.} \quad (14)$$

We have one such equation for each point on the image where s''_k and s'_j can be computed from the image gradients and $p^i = (x, y, 1)$ are the (projective) image coordinates of the point in Image 1. We solve for $\mathcal{T}_i^{jk}$ which combines the motion and camera parameters. *The coordinates of the corresponding points (x', y') and (x'', y'') are not required.*

Every pixel with a non-vanishing gradient contributes one linear equation to the 27 unknown parameters comprising $\mathcal{T}_i^{jk}$. However, note that the lines s', s'' (equations 12 and 13) are *parallel*. A configuration of a point in the first image and two *parallel* lines in Images 2 and 3 is a particular instance of a degenerate line configuration called a Linear Line Complex. The general solution for the LLC case is explored in [46]. In this particular case the system of equations can provide a linear solution to 21 of the parameters and the remaining 6 parameters of $\mathcal{T}_i^{jk}$ can be determined using quadratic admissibility constraints. We provide here a proof for this special case:

From equation (14) the coefficients of the terms $\mathcal{T}_i^{12}$ and the terms $\mathcal{T}_i^{21}$ are both $p^i I_x I_y$. Therefore the linear equations cannot be solved for the six terms ($\mathcal{T}_i^{12}$ and $\mathcal{T}_i^{21}$) individually but only for the three sums

$$\mathcal{T}_i^{12} + \mathcal{T}_i^{21}, i = 1...3$$

.

The sum $\mathcal{T}_i^{12} + \mathcal{T}_i^{21}$ provides one linear equation in $\mathcal{T}_i^{12}$ and $\mathcal{T}_i^{21}$. It is known that the matrices $\mathcal{T}_1^{jk}, \mathcal{T}_2^{jk}, \mathcal{T}_3^{jk}$ are of rank 2. This leads to a quadratic equation in $\mathcal{T}_i^{12}$ and $\mathcal{T}_i^{21}$. The quadratic and linear equations together result in two solutions for each pair $\mathcal{T}_i^{12}, \mathcal{T}_i^{21}, i = 1...3$ for a total of 8 discrete solutions. A unique solution is obtainable by applying further "admissibility" constraints as shown in [46].

This added complexity in finding the unique solution leads us to search for simpler models. We next consider the case of a calibrated camera with the rotation limited to small rotation angles.

3.3. The Small Rotation Model with Calibrated Cameras

The next model is defined for small-angle rotations with calibrated cameras. Assuming calibrated cameras equations (2) and (3) become:

$$\begin{array}{rcccl} p' & \cong & [\mathbf{R}'; t']\, P & \cong & \mathbf{R}'p + kt' \\ p'' & \cong & [\mathbf{R}''; t'']\, P & \cong & \mathbf{R}''p + kt'' \end{array} \quad (15)$$

i.e., $\tilde{A} = [\mathbf{R}'; t']$ and $\tilde{B} = [\mathbf{R}''; t'']$ and $k = \frac{1}{z}$ replaces ρ in equation (2) and (3).

If we also assume small angle rotations and can therefore make the approximations:

$$\cos(\theta) \approx 1, \quad \sin(\theta) \approx \theta$$

then the rotation matrices can be approximated as:

$$\begin{array}{rcl} R' & \approx & [I + [w']_x] \\ R'' & \approx & [I + [w'']_x] \end{array} \quad (16)$$

where w', w'' are the angular velocity vectors and $[\cdot]_x$ is the skew-symmetric matrix of vector products. Equations (2) and (3) now become:

$$\begin{array}{rcccl} p' & \cong & \tilde{\mathbf{A}}P & \cong & [I + [w']_x; t']\, P \\ p'' & \cong & \tilde{\mathbf{B}}P & \cong & [I + [w'']_x; t'']\, P. \end{array} \quad (17)$$

We can now proceed along two paths. The different results will give us some further insight into the problem. First we will substitute $\tilde{A} = [I + [w']_x; t']$ and $\tilde{B} = [I + [w'']_x; t'']$ directly into equation (8) and then write out the tensor explicitly:

$$T_1 = \left[\begin{array}{ccc} t'_1 - t''_1 & t'_1 w''_3 - t''_2 & -t'_1 w''_2 - t''_3 \\ t'_2 - t''_1 w'_3 & t'_2 w''_3 - t''_2 w'_3 & -t'_2 w''_2 - t''_3 w'_3 \\ t'_3 + t''_1 w'_2 & t'_3 w''_3 + t''_2 w'_2 & -t'_3 w''_2 + t''_3 w'_2 \end{array}\right] \quad (18)$$

$$T_2 = \begin{bmatrix} -t_1'w_3'' + t_1''w_3' & t_1' + t_2''w_3' & t_1'w_1'' + t_3''w_3' \\ -t_2'w_3'' - t_1'' & t_2' - t_2'' & t_2'w_1'' - t_3'' \\ -t_3'w_3'' - t_1''w_1' & t_3' - t_2''w_1' & t_3'w_1'' - t_3''w_1' \end{bmatrix} \quad (19)$$

$$T_3 = \begin{bmatrix} t_1'w_2'' - t_1''w_2' & -t_1'w_1'' - t_2''w_2' & t_1' - t_3''w_2' \\ t_2'w_2'' + t_1''w_1' & -t_2'w_1'' + t_2''w_1' & t_2' + t_3''w_1' \\ t_3'w_2'' - t_1'' & -t_3'w_1'' - t_2'' & t_3' - t_3'' \end{bmatrix} \quad (20)$$

where T_1, T_2 and T_3 are the Standard Correlation Slices [46]. Studying the tensor terms we notice that the terms are not linearly independent and the following seven linear equations hold:

$$\begin{array}{l} \mathcal{T}_1^{1,2} + \mathcal{T}_2^{1,1} + \mathcal{T}_1^{2,1} - \mathcal{T}_2^{2,2} = 0 \\ \mathcal{T}_2^{2,1} + \mathcal{T}_1^{2,2} + \mathcal{T}_2^{1,2} - \mathcal{T}_1^{1,1} = 0 \\ \mathcal{T}_3^{1,1} + \mathcal{T}_1^{1,3} + \mathcal{T}_1^{3,1} - \mathcal{T}_3^{3,3} = 0 \\ \mathcal{T}_1^{3,3} + \mathcal{T}_3^{3,1} + \mathcal{T}_3^{1,3} - \mathcal{T}_1^{1,1} = 0 \\ \mathcal{T}_2^{3,3} + \mathcal{T}_3^{3,2} + \mathcal{T}_3^{2,3} - \mathcal{T}_2^{2,2} = 0 \\ \mathcal{T}_3^{2,2} + \mathcal{T}_2^{2,3} + \mathcal{T}_2^{3,2} - \mathcal{T}_3^{3,3} = 0 \\ \mathcal{T}_2^{1,3} + \mathcal{T}_3^{2,1} + \mathcal{T}_1^{2,3} + \mathcal{T}_3^{1,2} + \mathcal{T}_1^{3,2} + \mathcal{T}_2^{3,1} = 0 \end{array} \quad (21)$$

Therefore, for the calibrated small rotation model, the 27-parameter Tensor Brightness Constraint reduces to a *20-parameter* constraint equation. Each point in the image gives one homogeneous equation resulting in a set of N homogeneous equations, where N is the number of image points. This can be written in matrix form as:

$$\mathbf{A}x = 0$$

where $\mathbf{A}$ is the $N \times 20$ estimation matrix and x is the 20 parameter vector of the unknown intermediate parameters. There is a unique non-trivial solution if the estimation matrix is of $rank = 19$. But as in the projective case, the system of homogeneous equations is degenerate and in this case the rank of the estimation matrix is 16 not 19.

In [39, 45] a model-based brightness constraint for the calibrated, small rotation model was derived in a different way. Expanding equation (17):

$$p' \quad \cong \quad [I + [w']_x]\, p + kt' \quad (22)$$

and taking the dot product of the left-hand-side and right-hand-side terms with s' (equation 12) results in the equation:

$$0 \quad = \quad s'^\top p + s'^\top [w']_x p + k s'^\top t' \quad (23)$$

This can be simplified to the form:

$$k s'^\top t' + (p \times s')^\top w' + I_t' \quad = \quad 0 \quad (24)$$

by noting that $s'^\top p = I_t'$ and $s'^\top [w']_x p = (p \times s')^\top w'$ where:

$$p \times s' = \begin{pmatrix} -I_y + y(I_t' - xI_x - yI_y) \\ I_x - x(I_t' - xI_x - yI_y) \\ xI_y - yI_x \end{pmatrix}.$$

In a similar manner, for the second motion:

$$k s''^\top t'' + (p \times s'')^\top w'' + I_t'' \quad = \quad 0 \quad (25)$$

where

$$p \times s'' = \begin{pmatrix} -I_y + y(I_t'' - xI_x - yI_y) \\ I_x - x(I_t'' - xI_x - yI_y) \\ xI_y - yI_x \end{pmatrix}.$$

Multiplying equations (24) by $s''^\top t''$ and equations (25) by $s'^\top t'$ and subtracting we obtain the following *24-parameter* model-based brightness constraint as a reduction of the tensor brightness constraint:

$$I_t'' s'^\top t' - I_t' s''^\top t'' + s'^\top [t' w''^\top](p \times s'') - s''^\top [t'' w'^\top](p \times s') = 0. \quad (26)$$

The resulting 24-parameter model (equation 26) is not as compact as the 20-parameter model we obtained by direct substitution of the small rotation approximation into the tensor brightness constraint equation. We find that the $24 \times N$ estimation matrix obtained using the 24-parameter model is again of rank=16. Due to the high degeneracy of the linear equations neither the 24-parameter model nor the 20-parameter model are convenient to work with.

3.4. The 15-Parameter, Small Motion Model

A unique solution is obtained when we reduce the motion model further to include infinitesimal motion using the model introduced by Longuet-Higgins & Prazdny [24]. The LH&P model assumes in addition to small rotation that $\frac{t_z'}{Z} \ll 1$. The motion field equations for the first camera motion are then:

$$u' = \frac{1}{z}(t_1' - xt_3') - w_3'y + w_2'(1 + x^2) - w_1'xy \quad (27)$$

$$v' = \frac{1}{z}(t_2' - yt_3') + w_3'x - w_1'(1 + y^2) + w_2'xy$$

By substituting equation (27) into the optic flow constraint equation:

$$u'I_x + v'I_y + I_t' = 0 \quad (28)$$

and rearranging the terms we obtain:

$$k s^\top t' + v^\top w' + I_t' = 0 \quad (29)$$

where $k = \frac{1}{z}$ denotes the inverse depth at each pixel location and where s, v are defined below:

$$s = \begin{pmatrix} I_x \\ I_y \\ -xI_x - yI_y \end{pmatrix} \quad (30)$$

and

$$v = p \times s = \begin{pmatrix} -I_y - y(xI_x + yI_y) \\ I_x + x(xI_x + yI_y) \\ xI_y - yI_x \end{pmatrix} \quad (31)$$

Note that v is a vector that can be computed from image brightness, and should not be confused with v' which is the y component of the image flow between Image 1 and Image 2. Equation (29) which was first derived in [30, 20] can also be derived directly from equation (24) by making the LH&P assumptions: $\frac{t'_z}{Z} \ll 1$, $\frac{w'_y x}{f} \ll 1$ and $\frac{w'_x y}{f} \ll 1$.

Similarly for the second motion:

$$ks^\top t'' + v^\top w'' + I''_t = 0 \tag{32}$$

By eliminating k from equations (29) and (32) we obtain the *15-parameter model-based brightness constraint*:

$$\boxed{I''_t s^T t' - I'_t s^T t'' + s^T[t' w''^T - t'' w'^T]v = 0} \tag{33}$$

Equation (33) is a key equation. There is one such equation for every point in the image. The unknowns are the motion parameters t', t'', w' and w''. The values I'_t, I''_t, s and v, can all be computed from image derivatives and the coordinates of the point in Image 1. Given the solution to the ego-motion parameters (to be described later), one can recover the dense depth map from equations (29) and (32).

Before showing how to solve equation (33) for both rotation and translation we will start with the simple case of pure translation. This case arises in practice when we can either assure pure translation or after we have rectified the images (perhaps by registering a common plane in the images as in [22, 21, 33]). It is also an important case from a theoretical point of view. We will show that there are problems in recovering the motion parameters in the case of collinear motion. If collinear motion causes a problem for the pure translation case, it will also create a problem for the more general translation and rotation case.

3.5. Pure Translation

The pure translational small-motion model takes the simplest form:

$$I''_t s^T t' - I'_t s^T t'' = 0 \tag{34}$$

We have one such equation for each image point and we can write it out in the matrix form:

$$\mathbf{A}t = 0$$

where:

$$t = \begin{pmatrix} t'_x & t'_y & t'_z & t''_x & t''_y & t''_z \end{pmatrix}^T$$

and $\mathbf{A}$ is an $N \times 6$ matrix with the $i'th$ row (corresponding to the $i'th$ pixel) given by:

$$\begin{pmatrix} I''_t s_{i1} & I''_t s_{i2} & I''_t s_{i3} & I'_t s_{i1} & I'_t s_{i2} & I'_t s_{i3} \end{pmatrix}$$

To avoid the trivial solution $t = 0$ we add the constraint $\|t\| = 1$. The least squares problem now maps to the problem of finding $\|t\| = 1$ that minimizes:

$$t^T \mathbf{A}^T \mathbf{A} t = 0$$

The solution is the eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to the smallest eigenvalue.

3.5.1. The Singularity of Collinear Motion

This method fails when the two motions are in the same (or opposite) directions. In the pure translation case

$$ks^\top t' + I'_t = 0 \tag{35}$$

and

$$ks^\top t'' + I''_t = 0 \tag{36}$$

If the second translation vector t'' is proportional to the translation vector t' then equation (36) is simply a scaled version of equation (35) adding no new information. The solution will therefore be ill conditioned when $t' \approx \beta t''$ for a scalar β. This is a drawback in many applications (e.g. 3D reconstruction from a monocular image sequence). We present in Section 6.1 some early research on ways to overcome this problem.

3.6. Solving for Translation and Rotation

In equation (33) the ego-motion parameters are embedded in a 15 parameter model: 6 translation parameters and the 9 outer-product terms $[t'w''^T - t''w'^T]$. Each pixel provides one linear equation, thus N pixels provide a system $\mathbf{A}c = 0$ where $\mathbf{A}$ is an $N \times 15$ measurement matrix and c is the 15-vector of unknown parameters. The least squares solution is the eigenvector of $\mathbf{A}^\top\mathbf{A}$ corresponding to the smallest eigenvalue. The solution is unique if the null space is one dimensional, i.e., the rank of $\mathbf{A}$ is 14. We will prove however, that the null space is at least two dimensional.

Proposition 1 *Let $\mathbf{A}$ be the $N \times 15$ measurement matrix associated with the homogeneous equation (33), where $N \gg 15$. Then the null space of the matrix $\mathbf{A}^\top\mathbf{A}$ is of dimension greater or equal to 2, i.e., Rank($\mathbf{A}$) ≤ 13.*

Proof:
Since

$$s^\top v = s^\top (p \times s) = 0$$

the vector

$$c_0 = (0,0,0,0,0,0,1,0,0,0,1,0,0,0,1)^T$$

is in the null space of $\mathbf{A}$.
The vector c_0 corresponds to:

$$t' = 0$$
$$t'' = 0$$
$$\left[t'w''^T - t''w'^T\right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore c_0 is clearly not an admissible solution, and therefore the null space of $\mathbf{A}$ includes two vectors: the vector c_0 and the true solution. It follows that the rank of $\mathbf{A}$ is at most 13. ∎

The method of solution is based on the following observation. Let b_0 be another vector of the null-space (for example, let c_0 and b_0 be the two eigenvectors of $\mathbf{A}^\top\mathbf{A}$ corresponding to the two smallest eigenvalues), then the desired solution vector b is a linear combination of the two:

$$b = b_0 + \alpha c_0 \quad (37)$$

We note that c_0 is in the null space of $\mathbf{A}$ regardless of noise in the data and will therefore correspond to an eigenvalue of zero, up to numerical precision. Theoretically, b_0 is also in the null space of $\mathbf{A}$ but only in the case of noise free data and exact motion model. Therefore b_0 is the eigenvector corresponding to the second smallest eigenvalue. We next show how to determine α.

3.6.1. Solution Using the Rank=2 Constraint.

In order to find α given c_0 and b_0 we enforce the constraint that:

$$Rank[t'w''^T - t''w'^T] = 2.$$

Clearly the choice of α will have no effect on the first 6 elements of the vector b. Let us arrange the last 9 elements of b, b_0 and c_0 into the corresponding 3×3 matrices $\mathbf{B}$, $\mathbf{B_0}$ and $\mathbf{C_0}$. We are now looking for an α such that:

$$Rank(\mathbf{B_0} - \alpha \mathbf{C_0}) = 2. \quad (38)$$

In our case $\mathbf{C_0}$ is the identity matrix so (38) becomes:

$$Rank(\mathbf{B_0} - \alpha \mathbf{I}) = 2.$$

and the solution for α is given simply by the eigenvalues of $\mathbf{B_0}$. Since $\mathbf{B_0}$ is a 3×3 matrix this results in up to three discrete solutions. We now prove that only one of those can be a valid solution.

Theorem 1 *The 15-parameter model has a unique solution. Let:*

$$\mathbf{B} = \left[t'w''^T - t''w'^T\right]. \quad (39)$$

Let:

$$\tilde{\mathbf{B}} = \left[t'w''^T - t''w'^T\right] + \alpha \mathbf{I}. \quad (40)$$

Then the equation:

$$\tilde{\mathbf{B}} = \left[t'\tilde{w}''^T - t''\tilde{w}'^T\right] \quad (41)$$

has a solution for $\tilde{w}'', \tilde{w}'$ given t', t'' iff $\alpha = 0$.

Proof:

The *if* part of the proof is trivial. For $\alpha = 0$, $\tilde{w}'' = w''$ and $\tilde{w}' = w'$ is a solution for (41). To prove the *only if* part, suppose there exist $\tilde{w}'', \tilde{w}'$ that is a solution for (41). Then:

$$\left[t'w''^T - t''w'^T\right] + \alpha \mathbf{I} = \left[t'\tilde{w}''^T - t''\tilde{w}'^T\right]. \quad (42)$$

Let q be a vector such that $q^T t' = 0$, $q^T t'' = 0$ and $q \neq \vec{0}$. Multiply left and right side of (42) by q^T to get:

$$q^T \left[t'w''^T\right] - q^T \left[t''w'^T\right] + q^T \alpha \mathbf{I} = q^T \left[t'\tilde{w}''^T\right] - q^T \left[t''\tilde{w}'^T\right]. \quad (43)$$

Since $q^T t' = 0$ and $q^T t'' = 0$ most of terms in (43) become zero and we are left with:

$$q^T \alpha \mathbf{I} = \vec{0}. \quad (44)$$

Since $q \neq \vec{0}$ then $\alpha = 0$. $\square$

3.6.2. The Algorithm for Finding Motion Parameters

Based on the previous arguments the algorithm for finding the motion parameters is as follows:

1. Compute the $N \times 15$ matrix $\mathbf{A}$ from equation (33).
2. Find the eigenvector corresponding to the second smallest eigenvalue of the matrix $\mathbf{A}^T\mathbf{A}$. This is b_0. (The vector c_0 corresponds to the smallest.)
3. The first 6 elements of b_0 are the translations, t',t''.
4. Arrange the last 9 elements of b_0 into a 3×3 matrix $\mathbf{B_0}$.
5. Construct three possible rank 2 matrices $\mathbf{B}_{i,i=1..3}$ from $\mathbf{B}_0$:

   $$\mathbf{B}_i = \mathbf{B}_0 - \alpha_i \mathbf{I} \quad (45)$$

   where $\alpha_{i,i=1..3}$ are the three eigenvalues of $\mathbf{B}_0$.
6. For each $\alpha_{i,i=1..3}$ solve:

   $$[t'w''^T - t''w'^T] = \mathbf{B},$$

   for w' and w'', given t' and t'' from step (3). This a set of 9 linear equations in the six unknowns (w',w'') and is solved using least squares.
7. From Theorem 1 only one of the three solutions is correct. We select the solution that best fits the data. For each of the three solutions use the 12 parameters t', t'', w' and w'' to form the 15 intermediate parameter vector b_i and then compute the error:

   $$E_i = b_i^T \mathbf{A}^T \mathbf{A} b_i. \quad (46)$$

   Of the three solutions select the solution which gives the smallest error.

4. Implementation Details

4.1. Computing the Depth, Smoothing and Interpolation

After recovering the camera motion (Section 3.6.2) we use equations (29) and (32) to compute depth at every point. Information is combined from both image pairs by minimizing the least squares error:

$$E = \min_{k} \arg \sum_{j=1}^{2} \left(ks^T t^j + v^T w^j + I_t^j\right)^2.$$

Here $j = 1$ and $j = 2$ denote values from the first and second image pairs respectively.

There are points in the image where the brightness gradients are close to zero, (and therefore $s^T t^j \simeq 0$) and the estimation of k will be ill-conditioned at those points. In order to overcome this problem we use a local region of support around the point of interest. It is simplest form we assume the depth is constant in the region and minimize:

$$E = \min_{k} \arg \sum_{x,y \in R} \sum_{j} \beta(x,y) |s^T t^j|^p \left(ks^T t^j + v^T w^j + I_t^j\right)^2 \quad (47)$$

where the windowing function $\beta(x, y)$ allows one to increase the weights of the points closer to the center of the region. The $|s^T t^j|^p$ term reduces the weight of points which have a small gradient or where the gradient is perpendicular to the camera motion since these cases are highly affected by noise. We used $p = 1$.

During the iteration process we typically used a region R of 7×7 to 11×11. For 'prettier' results, we reduced the region R to 1×1 in the last iteration, but added a very weak global smoothness term and performed multi-grid membrane interpolation[31]. The smoothness term stabilizes regions where there is no image gradient so very small regions of support can be used.

4.2. Iterative Refinement

The constant brightness constraint is a linearized form of the Sum Square Difference (SSD) criteria. The linear solution can be thought of as a single iteration of Newton's method applied to the problem. Iterative refinement is performed as follows:

1. Calculate motion (using equation 33).
2. Compute depth (using equation 47).
3. Using the depth and motion, warp Images 2 and 3 towards Image 1 (Section 4.3).
4. Compute new time derivatives I'_t and I''_t.
5. Compute a new motion and depth estimate.

In the ideal case, as the final result, the warped images should appear nearly identical to Image 1. One must be careful and not simply compute the incremental changes in the translation $\delta t'$ and $\delta t''$. As the images are warped closer together and the translation estimate approaches zero, the system of equations (33) will become ill conditioned. Furthermore, since the equations are homogeneous we must enforce a constraint such as $|t'|^2 + |t''|^2 = 1$ to avoid the trivial solution. We would not wish to apply such a constraint to $\delta t'$ and $\delta t''$. Thus, one must compute the full translation model (previous iteration plus the incremental change to the translation). This problem does not arise for the rotations: w' and w''. It is in fact convenient to warp the images using the best rotation estimate and then compute only the incremental improvement in the rotations: $\delta w'$ and $\delta w''$. For example we might chose to warp the images using a more exact rotation model than the linearized small rotation model.

The method for computing the full translation model and incremental rotation is based on carrying the previous translation (and depth) solutions (cf.[14]), described below. Let Ψ_1, Ψ_2 and Ψ_3 be the three images. Assume we have $\hat{k}$, $\hat{t}^j$, $\hat{w}^j$, from the previous iteration. (Here again we use j to stand for either $'$ or $''$.) The translation components of image motion $\hat{u}^j_t$ and $\hat{v}^j_t$, and the rotational components $\hat{u}^j_r$ and $\hat{v}^j_r$ can be computed using $\hat{k}$, $\hat{t}^j$ and $\hat{w}^j$ in equation (27). These are then used to warp images Ψ_2 to $\hat{\Psi}_2$ and Ψ_3 to $\hat{\Psi}_3$. After warping, the images satisfy the brightness constraint equation:

$$\begin{aligned} I_x du' + I_y dv' + \hat{I}'_t &= 0 \\ I_x du'' + I_y dv'' + \hat{I}''_t &= 0 \end{aligned} \tag{48}$$

where the temporal derivatives at each pixel are given by:

$$\begin{aligned} \hat{I}'_t &= \hat{\Psi}_2 - \Psi_1 \\ \hat{I}''_t &= \hat{\Psi}_3 - \Psi_1 \end{aligned} \tag{49}$$

and du^j, dv^j are the (still unknown) differences between computed image motions and the real image motions:

$$\begin{aligned} du^j &= u^j - \hat{u}^j_t - \hat{u}^j_r \\ dv^j &= v^j - \hat{v}^j_t - \hat{v}^j_r \end{aligned} \tag{50}$$

Let:

$$\alpha^j = I_x \hat{u}^j_t + I_y \hat{v}^j_t \tag{51}$$

which can also be written as:

$$\alpha^j = \hat{k} s^T \hat{t}^j \tag{52}$$

Substituting equations (51) and (50) in equation (48) we get:

$$\begin{aligned} I_x(u' - \hat{u}'_r) + I_y(v' - \hat{v}'_r) + (\hat{I}'_t - \alpha') &= 0 \\ I_x(u'' - \hat{u}''_r) + I_y(v'' - \hat{v}''_r) + (\hat{I}''_t - \alpha'') &= 0 \end{aligned} \tag{53}$$

Substituting equation (27) in equation (53) we get modified versions of the equations (29) and (32)

$$\begin{aligned} k s^T t' + v^T \delta w' + (I'_t - \alpha') &= 0 \\ k s^T t'' + v^T \delta w'' + (I''_t - \alpha'') &= 0 \end{aligned} \tag{54}$$

We start our first iteration with $\hat{k}$, $\hat{t}^j$, $\hat{w}^j$ all zero and therefore $\alpha = 0$ as well.

4.3. Image Warping

Given an estimate of the camera motion and the depth at every point, we can warp Image 2 towards Image 1. Image warping in general is described in [52]. Let I'_{old} be the original Image 2 and let I'_{new} be the warped image we are trying to create. We must first define functions $x_{old}(x, y, ...)$ and $y_{old}(x, y, ...)$ which, given image coordinates in the new image (x, y) and possibly some extra parameters (symbolized by , ...), return the coordinates of the corresponding point in the old image (x_{old}, y_{old}).

In our case the *extra parameters* are the camera motions t' and w', and the depth map at every point $k(x, y)$. The functions x_{old} and y_{old} depend on the motion model. In our implementation we used the LH&P small motion model (equations (27)). Therefore:

$$\begin{aligned} x_{old} &= x + u' \\ &= x + \tfrac{1}{z}(t'_1 - x t'_3) - w'_3 y + w'_2(1 + x^2) - w'_1 xy \end{aligned} \tag{55}$$

and

$$\begin{aligned} y_{old} &= y + v' \\ &= y + \tfrac{1}{z}(t'_2 - y t'_3) + w'_3 x - w'_1(1 + y^2) + w'_2 xy. \end{aligned} \tag{56}$$

After defining x_{old} and y_{old} we can compute the value for every pixel in I'_{new} according to the formula:

$$I'_{new}(x,y) = \begin{cases} I'_{old}(x_{old}(x,y,...), y_{old}(x,y,...)), \\ \qquad if(x_{old}, y_{old}) \in I'_{old} \\ 0, \qquad\qquad otherwise. \end{cases} \tag{57}$$

There are two points to note:

- The values of x_{old} and y_{old} are in general non-integer, so we use bilinear interpolation to compute the appropriate pixel values. Bilinear interpolation works better than nearest neighbor. The use of the more complex bicubic interpolation resulted in no noticeable improvement over bilinear interpolation.
- If the coordinates (x_{old},y_{old}) are outside the coordinates of the image then we use the value 0. These points should be marked as invalid and not used in subsequent motion and depth computation.

4.4. Coarse-to-Fine Processing

In order to deal with image motions larger than 1 pixel we use a Gaussian pyramid for coarse to fine processing [3, 6]. Each of the three images is filtered by an approximation to a Gaussian filter and subsampled to create an image of half the size (in each dimension). This operation is performed recursively to create a pyramid with four or five levels depending on the size of the original image. The following 5×5 kernel was used as the filter:

$$\frac{1}{256} \times \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}.$$

This filter is separable and can be implemented by convolving the rows and columns of the image, twice each, with the filter $f = (0.25,\ 0.5,\ 0.25)$.

Starting from the coarsest level of we perform a few iterations to compute motion and depth according to the scheme described in Section 4.2. The brightness derivatives (I_x, I_y, I'_t, I''_t) are computed using the subsampled images. After computing the motion and depth, we create a finer depth map from the coarse depth map using bilinear interpolation. We then use the interpolated depth map, and the motion estimates, as the starting values for the iterations at the finer level.

4.5. Field of View

In this section we will first demonstrate that there exists an ambiguity between rotation and translation in the general problem of determining camera motion from motion fields when the field of view is narrow. We will also note the errors in the estimation of the rotation if we use the wrong focal length value. Then we show how this ambiguity is observed in the shape of the cost function:

$$E(t', t'', w', w'') = b^\top \mathbf{A}^\top \mathbf{A} b. \tag{58}$$

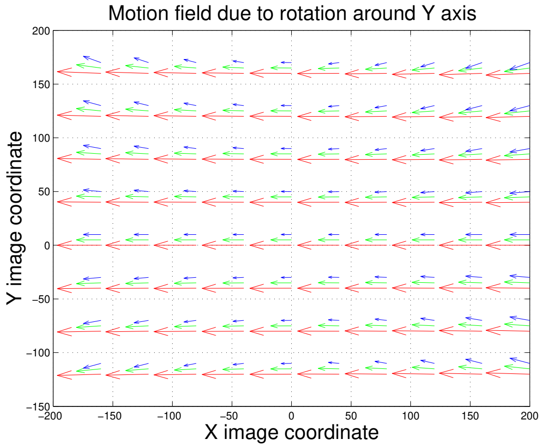


Fig. 5. *Motion flow field due to rotation around the Y axis for wide, medium and narrow fields of view. The arrows for the medium and wide have been offset in the Y direction for clarity. For the narrow field of view ($f = 800_{pixels}$, $FOV{=}30^o$) the flow field is large and almost parallel to the X axis. For wider fields of view ($f = 400$ and $f = 200$) the X component of the flow is smaller but there is a noticeable Y component.*

Finally we describe how to achieve stable and accurate motion estimates in the presence of rotation even without very wide fields of view.

4.5.1. Rotation-Translation Ambiguity from Motion Fields

It is well known that for a medium to narrow field of view, the motion field due to camera rotation around the Y axis, can be indistinguishable from the motion field due to some translation along the X axis and an appropriate depth surface. A similar ambiguity exists between rotations around the X axis and translations along the Y axis.

In Figure 5 we see that for medium-narrow fields of view (30^o) the flow due to rotation around the Y axis is parallel to the X axis. The length of the flow vectors due to rotation is slightly shorter, closer to the Y axis. A similar flow field can be produced by camera translation along the X axis with a surface that is curved so that points in the center of the image are more distant. There are motion flows that unambiguously indicate a translational component to the motion: motions fields that include a focus-of-expansion or show parallax effects. Nevertheless even these fields are ambiguous since a rotational component could be added to the motion, compensated for by a change in the translation and depth to produce the same motion field. In some cases the resulting depth map will include negative depths indicating some obvious error. This is often formulated as the *depth positive constraint*[13] which can be used to limit the range of motion ambiguity.

With wider fields of view (i.e. shorter focal lengths) the ambiguity disappears. The motion fields due to rotation cannot be modeled by translational motion. Figure 5 also shows the motion fields due to rotation with fields of view

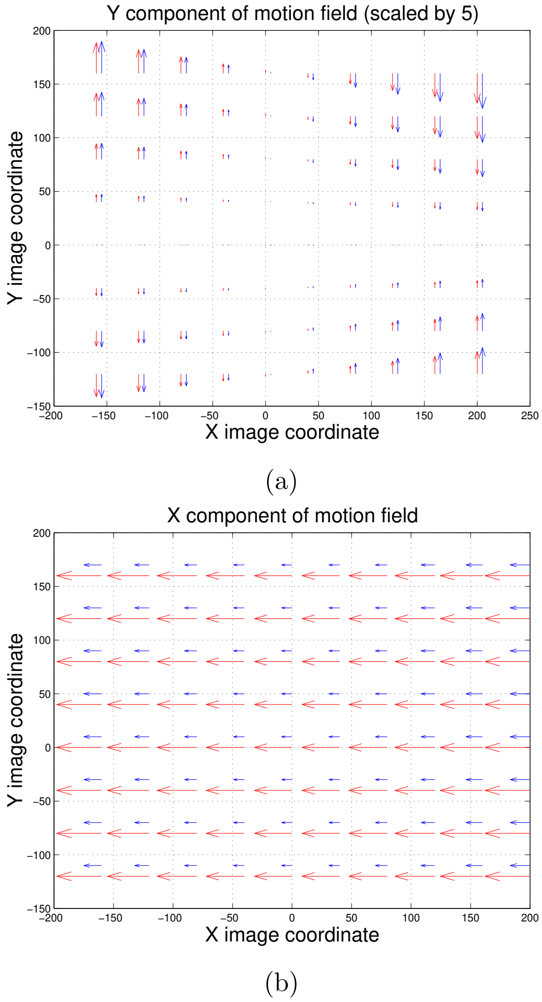


Fig. 6. *The X and Y components of the motion field due a rotation around the Y axis of -4.5^o with a focal length of 400_{pixels} for a rotation of -2.5^o with a focal length of 200_{pixels}. The arrows for the 200_{pixels} camera have been plotted offset right (a) and up (b). Note that the Y flow components are very similar.*

of 53^o and 90^o. The distribution of the Y component of the flow has the unique characteristics of rotational flow. For wide fields of view the Y component points towards the X axis on side of the image and away from the X axis on the other.

The motion field for a wide field of view is unambiguous if we know the focal length, or equivalently, the field of view (i.e. the camera's internal parameters are known). If our estimate of the focal length is incorrect then the motion and structure estimates which best fit the flow field will be incorrect. If the focal length estimate is smaller than the true focal length then the estimated angle of rotation will be smaller than the true rotation angle (in absolute values). Figure 6a shows the X and Y components of the motion field of a rotation of -4.5^o around the Y axis using a focal length of 400_{pixels}. Figure 6b shows the same measurements for a focal length of 200_{pixels} and a rotation of -2.5^o. The Y components are similar. The X components are very different but these can be accounted for by adjusting the translation and depth estimates.

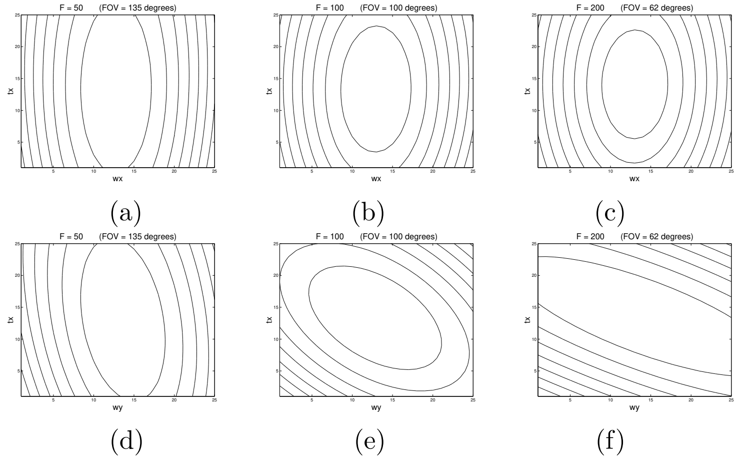


Fig. 7. *Contour plot of the cost function around the minimum for various fields of view. These simulation results were computed after the program converge to the final motion and depth estimates. For visualization only two of the 12 parameters were changed at one time. Note that for a narrower field of view (f) the cost function has the shape of a narrow diagonal valley. This indicates an ambiguity between translation along the X axis and rotations around the Y axis.*

4.5.2. Rotation-Translation Ambiguity in the 15-parameter Model

The rotation translation ambiguity can also be observed while using the 15-parameter model. Figure 7 show the shape of the cost function (equation 58) near the global minimum. These are simulation results for focal lengths of 50, 100 and 200. The image size was 320×240. It is not possible to plot the full 12 dimensional surface so the figures show only the shape of the function as we vary two motion parameters at a time. We show the effect of varying w'_x and t'_y where we expect to see ambiguity and the effect of varying w'_x and t'_x where no ambiguity is expected. In Figure 7f, the long diagonal valley shape of the cost function indicates the ambiguity of the motion estimate for narrower fields of view.

4.5.3. The Solution

As we have seen, in order to get reasonable rotation and translation estimates a wide field of view is required. The theoretical results of [27] and simulation results indicate that very wide fields of view are required (120^o or greater). These are hard to achieve with standard lenses. With narrower fields of view the iterative scheme described in Section 4.2 fails to converge. In previously published work [45] we managed to stabilize the results for small rotation angles by unknowingly biasing the results towards small rotation. This was done by setting the focal length parameter to 50 instead of around 600. For a 640×480 size image this implied a field of view of 150^o instead of the correct 55^o. As a result the rotation estimates were considerably smaller that the true values. This also resulted in errors in the translation and depth estimates.

It is important to note that while the magnitude of the angle estimates in [45] were too small, the estimated axis of rotation was correct. In fact the results show a linear relationship between true angle and estimated angle. That the rotation values were always smaller than the true values is consistent with the simulation results shown in Figure 6. This leads us to the following simple modification to the iterative scheme (Section 4.2):

1. For the estimation of motion use a focal length parameter $f << f_{true}$.
2. During the image warping stage use the correct focal length to rotate Images 2 and Image 3 towards Image 1.

This modification required no changes to the structure of the algorithm since the coarse-to-fine and iterative mechanisms were already in place. The results shown in Section 5.4 show that the modified algorithm gives accurate rotation estimates even for medium fields of view (55^o).

4.5.4. How does this affect the translation estimates?

How does using the wrong focal length affect the translation estimates? From equation (33) (and also (29), (32) and (47)) we see that the translations t' and t'' always appear in a dot product with s where:

$$s = \begin{pmatrix} I_x \\ I_y \\ -xI_x - yI_y \end{pmatrix}.$$

The first and second terms of s are not affected by the focal length estimate. Therefore there is no effect on the translation estimate in the X and Y direction, t_x and t_y. In the third term, x and y are supposed to be the normalized image coordinates. If we use the wrong focal length then the estimated Z translation (t_z) will be scaled accordingly.

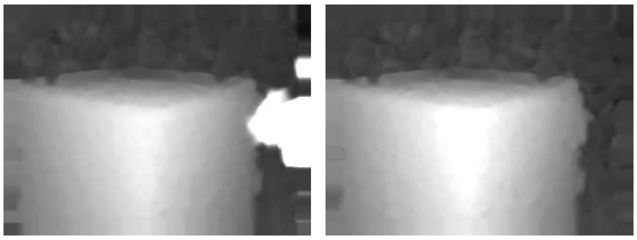
$$\frac{t_Z}{t_{Ztrue}} = \frac{f}{f_{true}}.$$

Using this simple relationship we can recover the true translation direction from the estimated one even if we use a focal length value which is much smaller than the true value.

4.6. Final Touches - Recompute the Depth Keeping the Motion Constant

Initially, during the first iterations and at the coarsest levels of the pyramid, the motion estimates are not accurate and cause errors in the depth estimates. Sometimes, particularly near the borders of the image, these errors are large enough to affect the new derivatives, I'_t and I''_t in such a way that at the next iteration, even when the motion estimates are more accurate, the depth values are still badly in error. This tends to be a local problem and does not affect the global motion estimates but it does create local 'holes' in the depth map.

To fix this problem, after computing the motion at the finest level, we go back down the pyramid and recompute the depth while keeping the motion estimate constant.



(a) (b)

Fig. 8. *3D reconstruction of an cube. An example of a local depth error (a) fixed by a second pass down the pyramid performing 'depth-only' computation (b). (see text)*

This is in effect becomes a three camera stereo computation. Figure 8 shows an example of the depth estimates before and after the second pass.

.

5. Experiments and Results

5.1. Overview

In this section we perform experiments which test various aspects of direct estimation of motion and structure. Experiments 5.2, and 5.3 deal with the pure translation case. In experiment 5.2, we test the accuracy of heading estimation for non-collinear motion. Experiment 5.3 tests a standard optic flow program on some of the same input images and compares the results with the 'direct methods'.

We then move on to case where the motion involves both translation and rotation. Experiment 5.4 revisits the image sequence used in Stein & Shashua [45]. We show that the improvements described in Section 4.5.3 greatly improve the rotation estimates. In Experiment 5.5 we repeat the heading estimation experiments but this time the motion includes some rotation. We also show how not taking into account even small rotations ($\sim 0.5^o$) leads to large errors in heading estimates. In Experiment 5.6 we test the accuracy of Euclidean reconstruction using the simple image of a cube. We measure whether the right-angles between the cube faces are correctly recovered.

In all the preceding experiments the camera was mounted on a motion stage so that ground truth motion could be known accurately. In Experiment 5.7 we repeat the Euclidean reconstruction experiments but this time with a hand-held camera. We show that if we neglect the rotation the shape recovery is noisy. We also show that if we neglect radial lens distortion then the reconstruction is qualitatively good but the Euclidean measurements, such as angles, are less accurate. This agrees with the observations regarding lens distortion and the trilinear tensor in [43]. Experiments 5.8 and 5.9 show results with simple outdoor scenes.

5.2. Accuracy of Heading Estimates for Pure Translation

In this experiment we measure the accuracy of the estimation of the camera motion direction (heading direc-

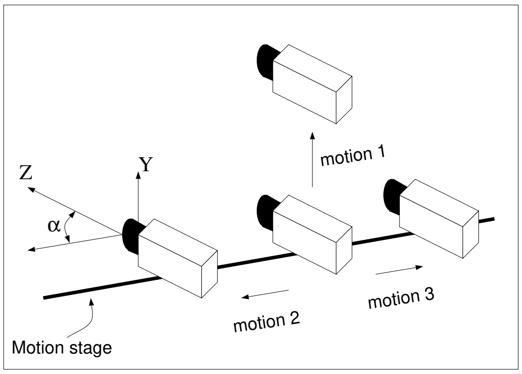
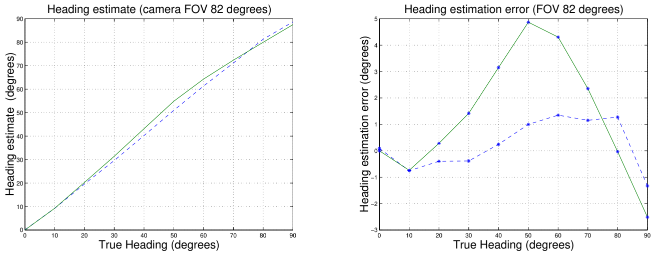


Fig. 9. *Schematic diagram of the pure translation experiment. Camera heading is set by the angle α. Motion 2 and Motion 3 are collinear.*



(a) (b)

Legend:	
——	No radial distortion correction
- - -	Images corrected for radial distortion

Fig. 10. *(a) Heading estimate from a translating camera. Pure translation was assumed. The 4.9_{mm} lens gives a 82^o FOV along the X axis. (b) Difference between heading estimate and motion stage reading ("True Heading").*

tion) in the case of pure translation. We also gauge the effect of the non-linear lens distortion.

5.2.1. Experimental Procedure:

The camera (Pulnix TM9701) was mounted on a motion stage with three degrees of computer controlled motion: horizontal translation, vertical translation and rotation around the camera's Y axis (Fig. 9). The camera lens was (4.9mm lens / 82^o FOV).

The camera Y axis was aligned with vertical axis of the stage. Initially the camera was positioned so that the optic axis was aligned with the horizontal axis of the translation stage. The accuracy of the alignment was $\pm 1^o$. The camera was rotated α^o and an image captured. Then the camera was translated vertically $12.5mm$ and an image captured. Finally the camera was translated horizontally $10mm$ and a third image captured. Thus we captured an image triplet where the camera motion was pure translation with one of the motions vertical and the second at a heading of α^o. This was repeated for angles $\alpha = 0^o, 10^o, \ldots, 90^o$.

5.2.2. Results:

Motion and depth estimation was performed with and without lens distortion correction. Figure 10 shows the heading estimates. The heading estimates are within a few degrees. Notice that when the true heading was inside the FOV the errors were less than 1^o. When the true heading was outside the FOV the errors increased to $1-2^o$. This is a case of a wide angle lens. The errors were considerably larger when lens distortion was not taken into account (solid line in Figure 10b).

5.3. Comparison with Optic Flow Techniques

After computing the motion and depth from an image triplet we can compute the motion flow using the motion equations (equation 27). In this experiment we compare the optic flow estimates obtained in this manner with the flow estimates computed directly from two of the images using an 'industry standard' optic flow program. The program is based on code by Bergen & Hingorani of the Sarnoff Corp [5, 4]. We use two sets of images which have been chosen to be particularly difficult for optic flow programs.

5.3.1. Experimental Procedure

Figure 3 shows three images of a real scene: two paper cylinders, a plaster bust and a black metal bar in front of a background of vertical stripes. The two motions are pure translation, vertical and horizontal, parallel to the image plane. Figure 3a shows the recovered depth map for the sequence. The results are qualitatively correct. In Figure 3b the texture mapping was removed for clarity. The difficulties in the scene are explained in the introduction (Section 1) and expanded upon in the results section below.

From the estimated motion and depth map, the motion filed was computed using equations (equation 27). The flow field was also computed using the optic flow program and the results compared.

This experiment was repeated for a second set of images shown in Figure 14a,b,c. The main difficulty here is that most of the gradients are smooth and have low contrast. The strongest edges appear at the occluding contour of the bust and these edges do not represent real features.

5.3.2. Results

Figure 11a shows the estimated flow computed from the recovered depth and motion. The camera motion was horizontal parallel to the image plane. The flow vectors have a zero Y component. Figure 11b shows the magnitude of the flow vectors. Figure 11c shows the flow estimated from two images using an optic flow program. Near the vertical bars on the left the flow vectors have a strong Y component. This is due to the aperture problem. Even a large window (aperture) will see image gradients in only one direction giving no constraint on the Y component of the flow.

On the cylinder to the lower right of the bust the magnitude of the flow vectors is too small. This is due to

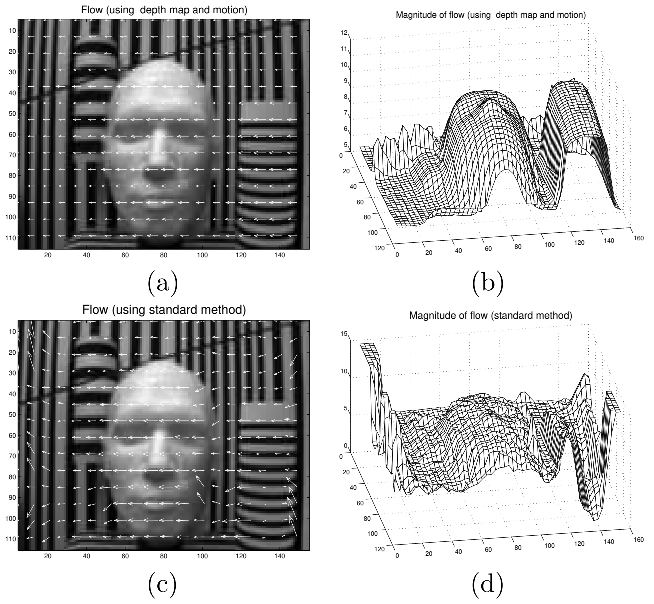


Fig. 11. *Optic flow computed for Figure 1. (a) Correct optic flow computed using depth map and recovered motion. The camera motion was horizontal parallel to the image plane. The flow vectors have a zero Y component. (c) Optic flow computed using code from Bergen & Hingorani. Near the vertical bars on the left the flow vectors have a strong Y component. This is due to the aperture problem. Even a large window (aperture) will see image gradients in only one direction giving no constraint on the Y component of the flow.*

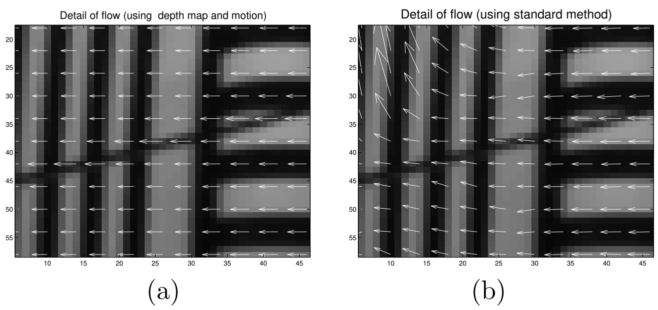


Fig. 12. *(a) Detail from the top left corner of Figure 11a. (b) Detail from the top left corner of Figure 11c. Along the vertical bars there is a strong Y component but this is to be expected due to the aperture problem. Lower down, near the intersections (in the image) of the diagonal line and the vertical bars the flow has a small Y component. This error occurs because the program 'tracks' the intersection point as if it were a real 'feature' point, but the lines do not intersect in space. Although smaller in magnitude this is the more significant error because the program will also give a high confidence to this value.*

the fact that the edges were horizontal and parallel to the horizontal epipolar lines. This means that the depth estimates for theses points are unreliable (Figure 11d). This problem does not occur in Figure 11b because two motions one horizontal and one vertical were used to compute the original depth map.

An enlarged detail of one of the more difficult regions is shown Figure 12. A critical error occurs near the intersections (in the image) of the diagonal line and the vertical bars.

The results for the second set of images are shown in Figure 13. Errors in flow direction can be seen near the edges of the bust. The most significant errors are errors

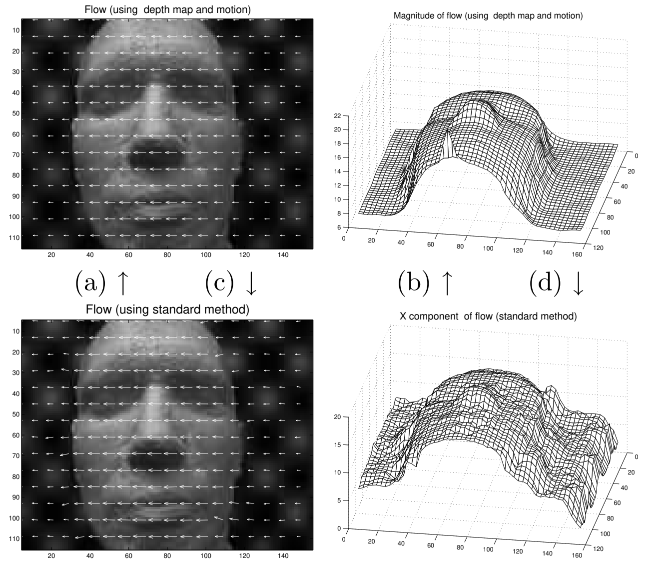


Fig. 13. *Optic flow computed for Fig. 14. (a) Correct optic flow computed using depth map and recovered motion. The camera motion was horizontal parallel to the image plane. The flow vectors have a zero Y component. (c) Optic flow computed using code from Bergen & Hingorani. Errors in flow direction can be seen near the edges of the bust. The most significant errors are errors in magnitude of the flow in the background areas. These can been seen clearly by comparing the mesh plots.*

in magnitude of the flow in the background areas. These can been seen clearly by comparing the mesh plots.

This experiment was intended to highlight the problems with optic flow techniques and how they are overcome by using direct methods. While this was not a simple 'straw man' and the optic flow program was one of the best available, a full system based on optic flow would have more components intended to overcome these problems. The optic program would compute a confidence ellipse for each flow vector. These estimates would then be combined using robust methods to reject outliers during the motion computation. After the camera motion was estimated the flow would be recomputed using the epipolar constraint and the new flow would be used to compute the depth. Probably three views would be used with different camera motions. Building such a worthy competitor and fine tuning it is beyond the scope of this paper, but this description does give an idea of the complexity of such a system.

5.4. Translation and Rotation (Reanalysis of Data from Stein & Shashua, CVPR97)

In this experiment we test the modified algorithm for recovering both translation and rotation (Section 4.5.3) on the same data set used by Stein & Shashua [45].

5.4.1. Experimental Procedure:

The images used by [45] were taken with an 8.5mm lens in the following way: The camera was translated first vertically (10mm) and then horizontally (5mm) to the right. At this third position the camera was rotated to various angles ranging from -4.0^o to 1.0^o. The depth

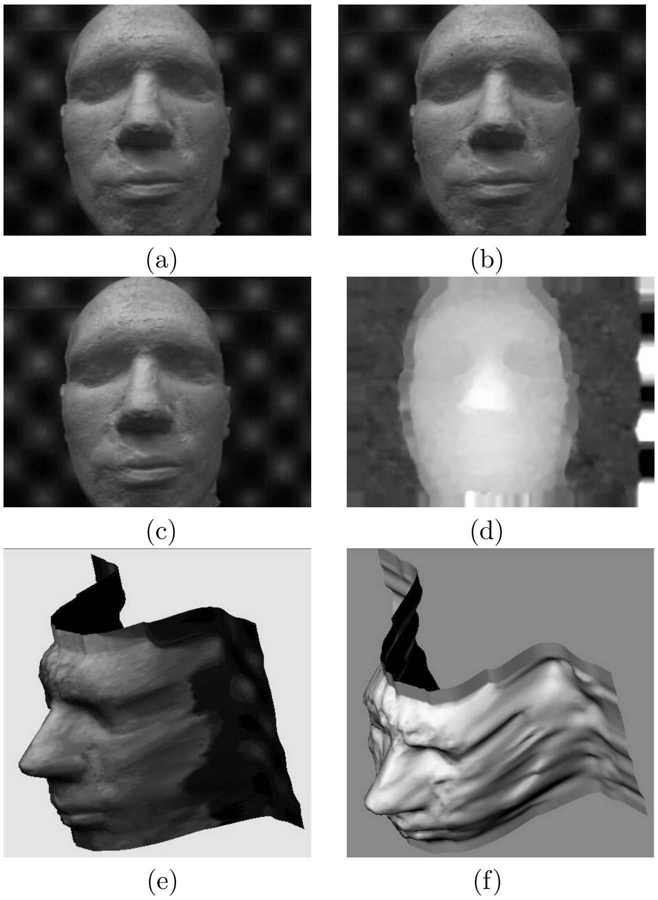


Fig. 14. *The three input images (a,b,d) used for 3D reconstruction. Motion ($a \rightarrow b$) was a horizontal translation with a -1.8^o rotation around the Y axis. Motion ($a \rightarrow c$) was a vertical translation. This is a challenging set of images because the the texture is smooth with low contrast. The strongest edge 'features' appear along the occluding contour of the head but this edge does not in fact correspond to a real feature in the world. (d) The estimated depth map. The estimated rotation was -1.62^o. There are of course errors around the boundary of the image where there is no overlap between the images. (e), (f) 3D rendering of depth map in (d)*

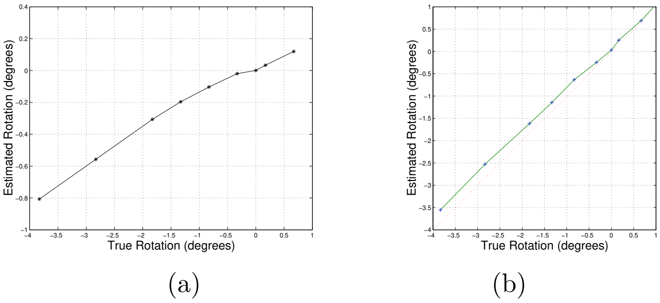


Fig. 15. *Estimated rotation as a function of true rotation. (a) Old results: Although there is the correct linear relationship there also appears to be a significant scale error (note Y axis). (b) New results - the rotation estimates are correct.*

in the scene ranged from $170mm$ to $400mm$.

Figs. 14a,b,c show three of the input images in the sequence. The flow due to a rotation of -1.8^o in Fig. 14b is much greater than the flow due to the translation.

5.4.2. Results:

Fig. 14d shows the recovered depth map using images in Figs. 14a,b,c. Figs. 14e,f show the 3D rendering of the depth map.

Fig. 15 shows the recovered rotation estimates for true rotations ranging from -4.0^o to 0.8^o. Using the new algorithm the rotation estimates are within 5% of the correct value. There is no scaling error as was observed using the old method.

The new algorithm also has a wider range of convergence. The original images were not available for larger rotations but a similar setup yielded good results for rotation for the range of -5.0^o to 5.0^o. Since rotations can create large amounts of image flow, key to the successful convergence are strong signals with low spatial frequency such as the white head on a darker background. This leads to convergence at the coarse level which is then used as the starting point for motion estimates at finer levels. Strong repetitive high frequency patterns such as checker board patterns are more difficult.

5.5. Motion Heading Estimation with Rotation

The aim of this experiment is to test the motion estimation algorithm over a wide variety of translation directions when the motion included some rotation. In particular, we test whether the algorithm works when the direction of motion is inside and outside the FOV. We also find the error created by neglecting even small amounts of rotation.

5.5.1. Experimental Procedure:

The camera setup is similar to that of experiment 1 (Section 5.2). The camera is rotated to a particular heading direction ($0^o, 10^o, ..., 90^o$). The camera was translated vertically and then horizontally. After the horizontal translation the camera was rotated $\pm 0.5^o$ and $\pm 1.0^o$. Therefore, for each image heading, we have an image triplet which is translation only and then triplets where the second motion also includes some small rotations. The rotation axis of the stage passed within $5mm$ of the camera center of projection thus rotations of 1.0^o produce translations of less that $0.1mm$. Figure 16 shows input images where the horizontal translation heading was 60^o to the camera optic axis.

5.5.2. Results:

Depth and motion estimates were computed either assuming pure translation or allowing possible rotation. The results for various motion and motion-assumption combinations are shown in Figure 17, Figure 18 and Figure 19. We notice that for even a small rotation (0.5^o) the heading estimate is off by over 10^o if we do not take the rotation into account. When we estimate both rotation and translation the motion estimates are good (RMS error 2.1^o) although not as good as those obtained from a purely translating camera under the pure translation assumption (RMS error 1.44^o).

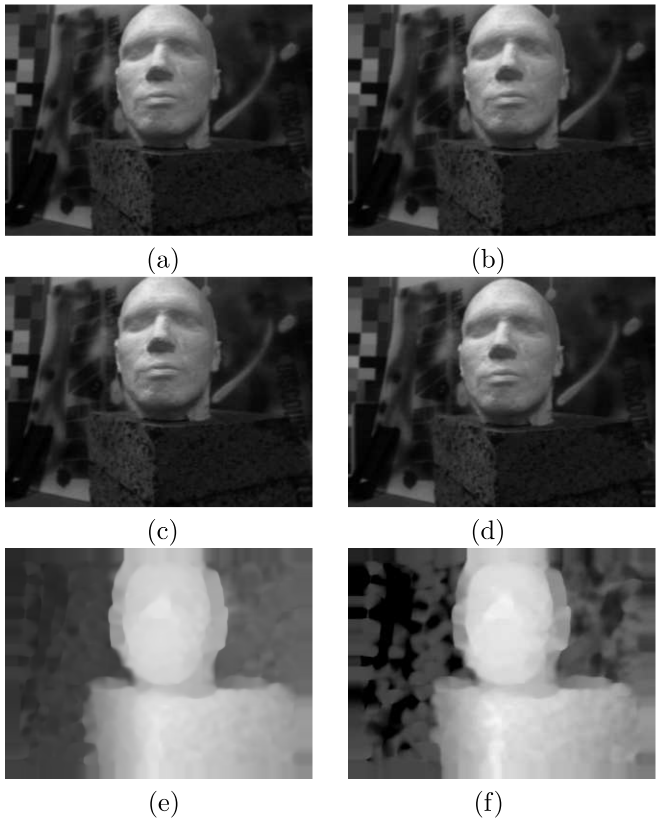


Fig. 16. *Four of the input images used in the experiment to test motion heading estimation in the presence of rotation. Motion (a) to (d) is vertical motion. Motion (a) to (b) is horizontal motion forward and to the left (60^o from the camera optic axis). Motion (a) to (c) has the same translation as (a) to (b) but also with a rotation of 1.0^o. In this case the flow due to rotation of 1.0 cancels out the flow due to translation in the area of the face and more than cancels out the flow in regions of greater depth. (e) Depth map estimated after computing both translation and rotation. The second camera motion included -1.0^o rotation. (f) Depth map estimated assuming pure translation when the second camera motion included 0.5^o rotation. Notice the depth errors in the background.(Lens 8.5mm lens / 52^o FOV)*

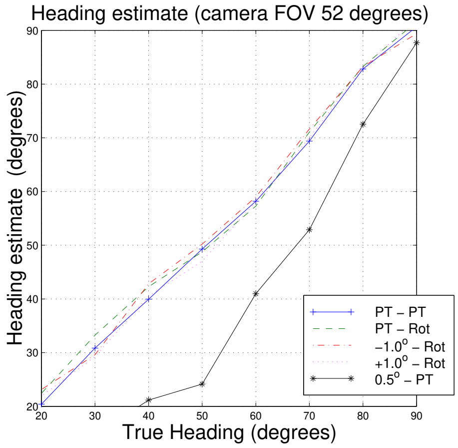


Fig. 17. *Heading estimates from a moving camera. (PT - PT) Both true motion motion model was pure translation. (0.5^o - PT) True motion included 0.5^o and motion model was pure translation. Note that even a small rotation, such as 0.5^o, is enough to create large heading estimation errors if rotation is not taken into account. (PT-Rot, $+1.0^o$ - Rot, -1.0^o- Rot) Motion model included rotation Actual rotations were $0^o, 1.0^o, -1.0^o$. (8.5_{mm} lens / 52^o FOV)*

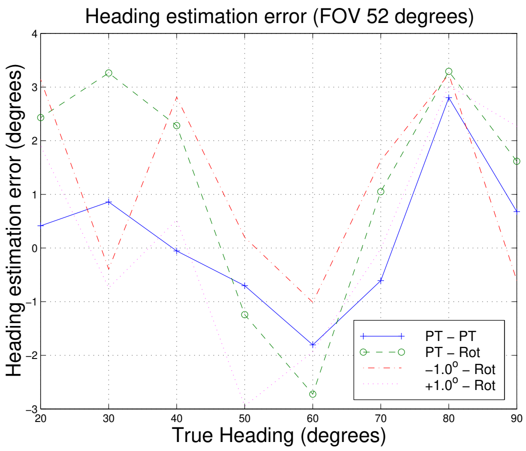


Fig. 18. *Heading estimate error: the difference between heading estimate and motion stage reading. Even in the presence of camera rotation ($+1.0^o$ - Rot, -1.0^o- Rot) the heading estimation errors are small (RMS error 2.1^o) although not as good as obtained from purely translating camera under the pure translation assumption (PT-PT) where the RMS error was 1.44^o. (8.5mm lens / 52^o FOV)*

Figure 16e shows the depth map estimated from the images in Figure 16a), Figure 16c and Figure 16d allowing for both translation and rotation. The depth map is qualitatively correct even though the second camera motion included -1.0^o rotation. Figure 16f shows the depth map estimated assuming pure translation when in fact the second camera motion included 0.5^o rotation.

5.6. Euclidean Structure Estimation in the Presence of Rotation.

In this section we evaluate the Euclidean structure estimation when the motion includes rotation. We use a simple scene, a cube, in front of a planar background. We perform the 3D reconstruction of the scene. We then measure whether the right-angles between the cube faces are correctly recovered.

5.6.1. Experimental Procedure

Figure 20 shows the simple scene of a textured cube in front of flat, low-contrast, background. The camera (8.5*mm* lens / 52^o FOV) was translated vertically and then horizontally parallel to the image plane. The camera was then rotated to $+2.0^o$ and -1.0^o from its original heading.

Since we were interested in the 3D structure and not just motion estimates, after estimating both structure and motion we went back down the pyramid and recomputed the structure while keeping the motion estimates constant (Section 4.6). The depth estimates before and after the second pass are shown in Figure 20d and Figure 20f respectively. In Figure 20d there is a "hole" on the right of the depth map. This is due to the appearance of a white patch near the border of the image. As with occlusions this sudden appearance violates the constant brightness assumption. The situation is fixed by the second pass. In general points near the edge of the depth map are unre-

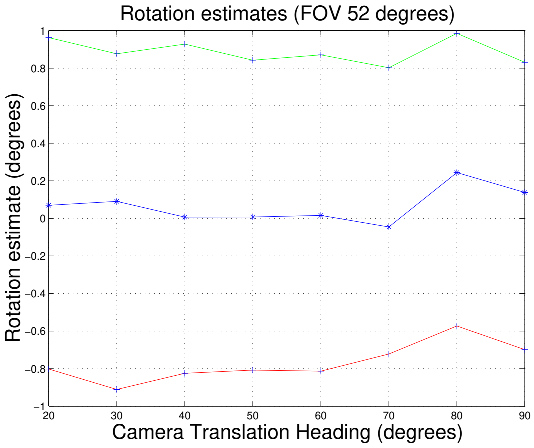


Fig. 19. *Rotation estimates obtained from a camera translating in different translation directions with some added rotation. Actual rotations were* $0^o, 1.0^o, -1.0^o$. *(camera with* 8.5_{mm} *lens /* 52^o *FOV)*

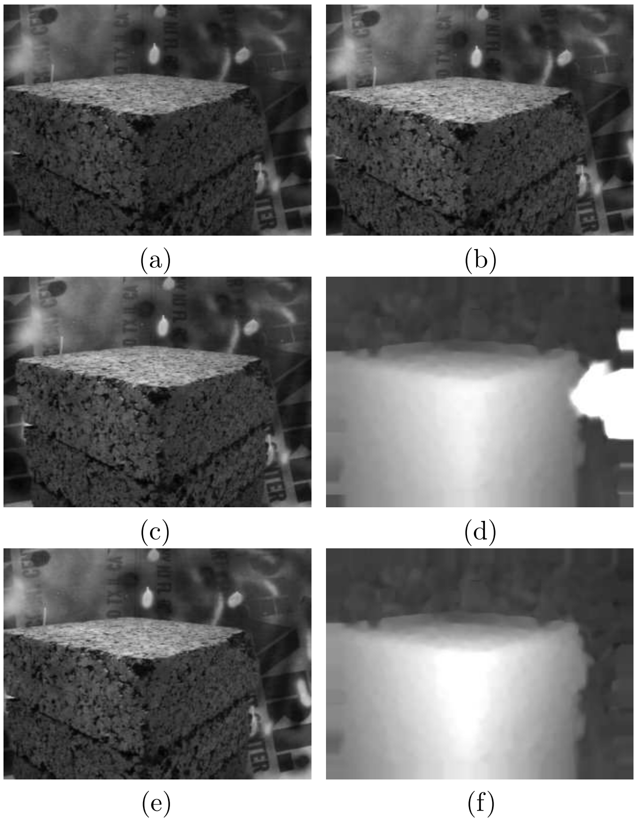


Fig. 20. *A simple scene used to test 3D Euclidean reconstruction: a cube constructed from cork blocks in front of a flat background. Motion (a) to (b) was vertical. Motion (a) to (c) was sideways translation with* -1.0^o *rotation. Motion (a) to (e) was same sideways translation but with* $+2.0^o$ *rotation. (d) initial depth map from images (a), (b) and (c). Note how the disappearance of a white blob in the background near the right edge of the image creates an error in one region of the depth map. (f) After second pass of depth estimation the "hole" is fixed. (see text)*

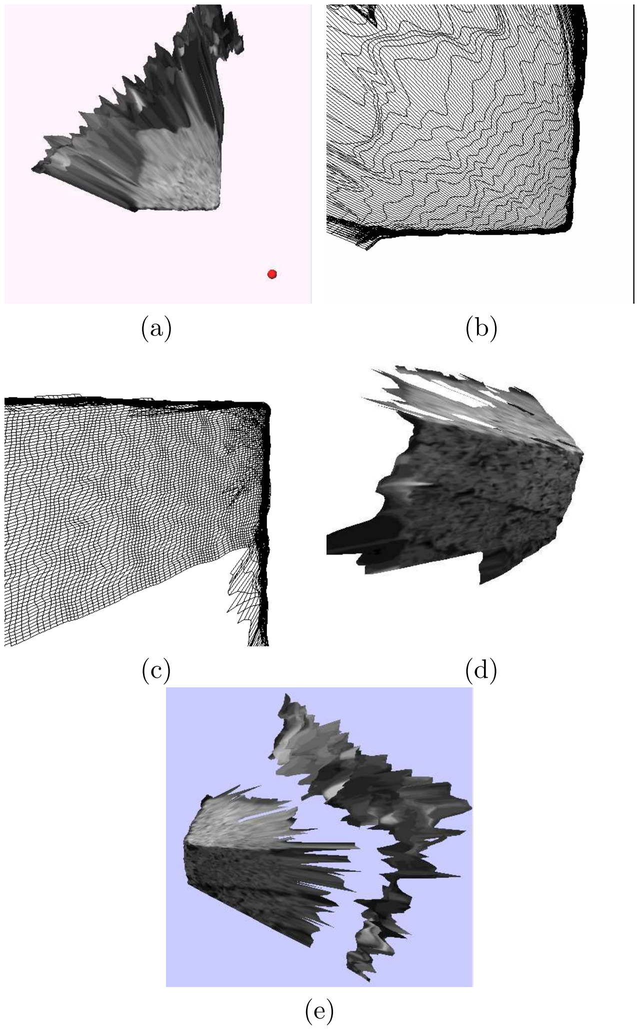


Fig. 21. *3D rendering of the recovered scene. The camera motion included* -1.0^o *rotation. (a) Overhead view. The estimated camera position is marked by the sphere in bottom right-hand corner. (b) An enlarged view of the corner of the cube. The angle between the faces of the cube is estimated* 95^o *instead of* 90^o. *(c) Side view of the segmented cube from up and to the side. (d) enlarged wire-frame side view of the cube. The estimated angle between the top face of the cube and the right face is* 90^o. *Another view of the scene showing the foreground/background segmentation. The depth estimates of the back panel are noisy because they are further away from the camera and because the texture does not have strong gradients.*

liable.

5.6.2. Results:

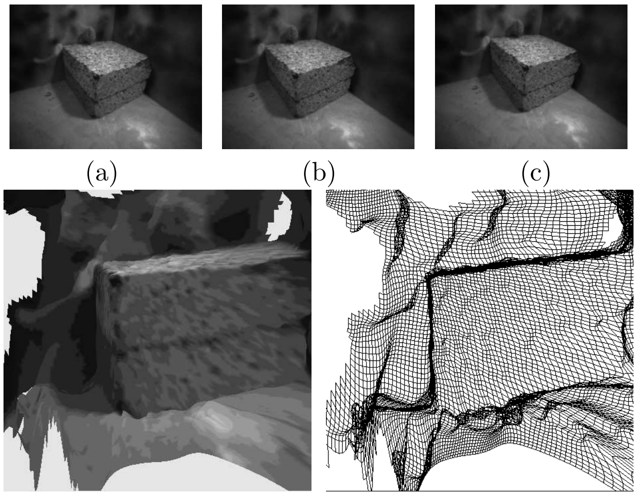
The rotation estimates were 1.68^o and -0.67^o for true rotations of $+2.0^o$ and -1.0^o respectively. The 3D rendering of the cube is shown in Figures 21 and 22. The wire-frame rendering of the overhead views (Figures 21b and 22b) show the recovered angle between the cube faces. The estimated angles were 95^o and 86^o respectively. Figure 21d shows a side view of the cube. The estimated angle between the top and side face of the cube is 90^o.

5.7. Euclidean Structure Estimation Using a Hand-held Camera

In previous experiments the camera was mounted on a motion stage so that ground truth motion values were available. In this experiment the camera was hand-held to show that the system can deal with the more natural situation. Simple objects were used to create the scene so that it would be easier to evaluate the shape reconstruc-

(a) (b)

Fig. 22. *3D rendering of the recovered cube. The camera motion included* $+2.0^o$ *rotation. The cube has been segmented from the background using depth based foreground/background segmentation. (a) View from up and to the side. (b) Wire-frame rendering of the overhead view. From the overhead view the estimated angle between the cube faces is* 86^o *instead of* 90^o



(a) (b) (c)

(d) (e)

Fig. 23. *A simple scene used to test 3D Euclidean reconstruction. Camera was hand-held so exact motion between images (a), (b) and (c) is not known. (d) Depth reconstruction allowing for both rotation and translation with radial distortion correction. (e) Assumes pure translation. (f) Allows for both rotation and translation but no radial distortion correction. The cube and it's surrounding are recovered well while textureless areas near the boundaries have large errors. (d),(e) 3D renderings of the recovered shape. The motion estimation allowed for rotation and translation and the input images were preprocessed to correct for lens distortion. The estimated angle between cube faces is* 96^o *instead of* 90^o.

tion.

5.7.1. Experimental Procedure

A progressive scan camera (Pulnix TMC9701) with a wide angle lens (4.9mm lens / 82^o FOV) was hooked up directly to the frame grabber of an SGI Indy workstation. The camera was handheld with no tripod and an image sequence of 30 frames was captured. An effort was made to keep the camera rotation down to a minimum.

From the image sequence three images were selected which seemed to give two distinct motion directions. Figures 23a,b,c show the three input images of the cube.

Table 1. *Rotation estimates (in degrees) from the images in Figure 23. The camera was hand-held so ground truth values of the motion are not known.*

	w_x	w_y	w_z
Motion 1	-0.7809	0.2262	-0.5520
Motion 2	-0.0538	0.2684	1.4246

5.7.2. Results

The motion and depth were estimated for three different cases. First the motion and depth were estimated allowing for both rotation and translation and taking into account the radial distortion. In the second case motion was estimated allowing for both rotation and translation but neglecting radial distortion. The resulting depth map is qualitatively the same but the measured angles between the cubes faces are larger and further from the true value of 90^o (Fig. 24c,d). In the third case assumed pure translation the resulting depth map again appears to be qualitatively the same but the 3D renderings (Figure 24e,f) clearly show that the results are more noisy and further from true Euclidean reconstruction.

Errors in depth reconstruction when we neglect the rotation are to be expected according to [8]. They show that for small total rotation angles ($|w|$) the error in depth due to an error in rotation ($|\delta w|$) is given by:

$$\frac{\delta Z}{Z} \approx \left(\frac{f}{|u|}\right)(|\delta w|)$$

where f is the focal length, $|u|$ is the magnitude of the image motion. In our case by assuming pure translation we get a rotation error of 1.45^o or $0.025rad$ (Table 1). The average image motions were on the order of $15pixels$ and the focal length $f \approx 180$. So we expect:

$$\frac{\delta Z}{Z} \approx 0.3.$$

[8] refers to depth from image motion. The errors we observe are not so large but are definitely noticeable. We also note that even if we neglect the rotation none of the depth estimates were negative so a *depth is positive* constraint, as suggested by [13], would not help in this example.

5.8. Outdoor Scenes - Church Wall

In this experiment we tested the algorithm on outdoor scenes. These experiments are important because they show that the photometric constraints can be used in uncontrolled lighting conditions. Figures 25a,b,c show three images of a small part of a church wall. The camera was Sony Hi8 camcorder with the lens open as wide as possible (FOV 40^o). The camera was mounted on a tripod. Moving the camera horizontally ($a \rightarrow b$) was quite smooth but moving the camera up and down included about 0.5^o rotation. Figure 25d shows the recovered depth map which can be seen to be qualitatively correct. The Euclidean 3D renderings of the depth map are shown in Figure 26.

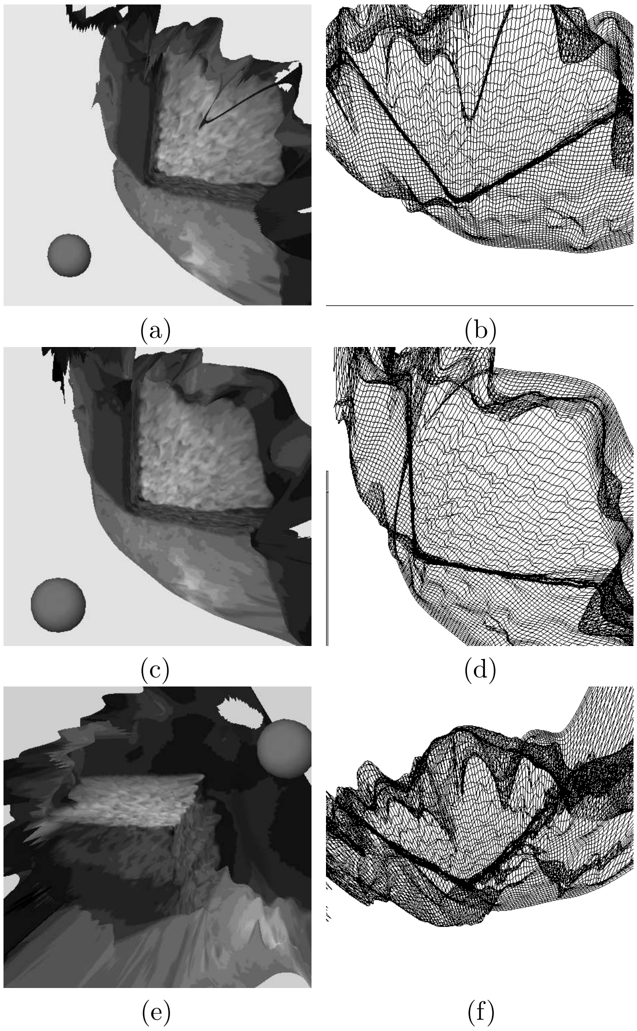


Fig. 24. *Top view of the cube scene in Figure 23. (a),(b) Motion and depth was estimated allowing for rotation. The images were preprocessed for radial distortion correction. The estimated angle between cube faces was* 97^o*. (c),(d) No radial distortion correction. The estimated angle between cube faces was* 101^o*. (e),(f) Motion was estimated assuming pure translation. There are many errors in the depth map and the angles estimate is* 105^o*.*

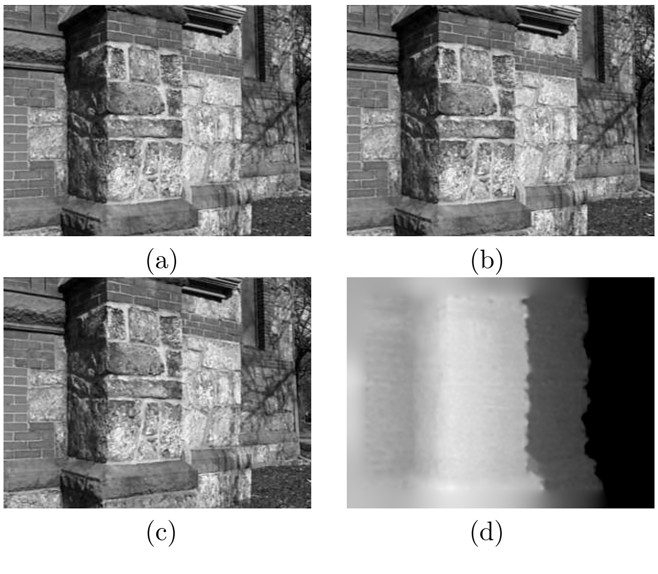


Fig. 25. *Detail of church wall: Three input views and resulting depth map. (Sony Hi8 camcorder, FOV* 40^o*)*

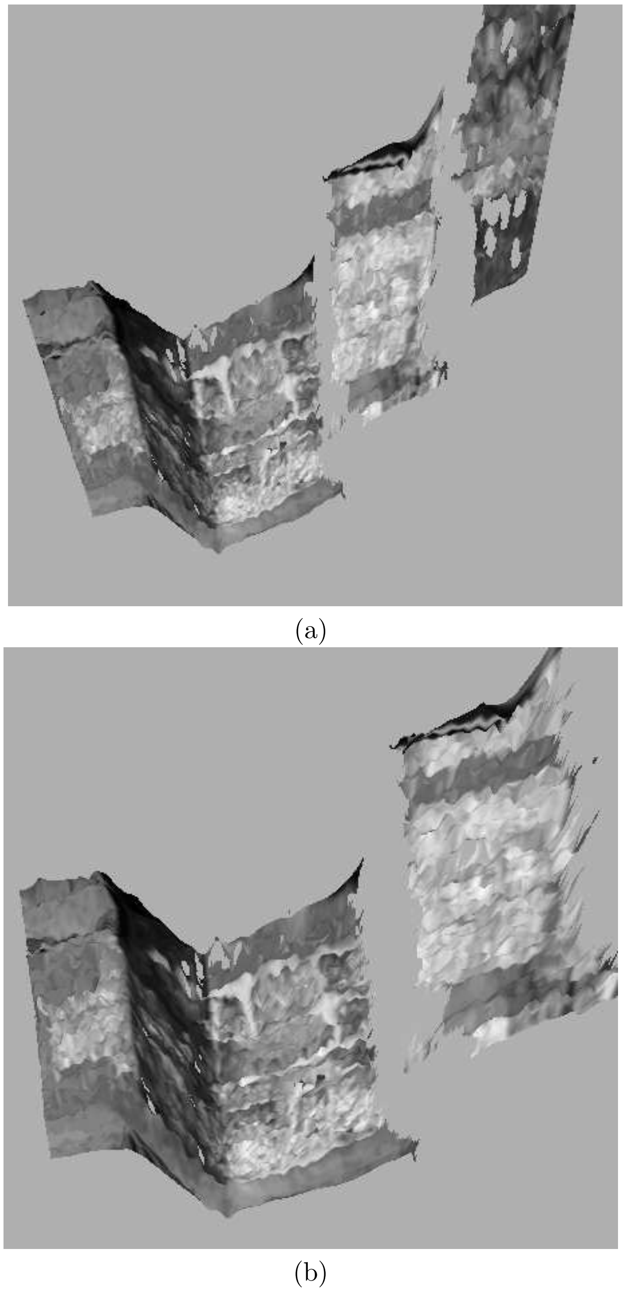


Fig. 26. *3D Euclidean renderings of the depth map in Figure 25d.*

5.9. Outdoor Scene 2 - Side Entrance

Figure 27 shows three views of the side entrance to a building. The camera was a progressive scan camera (Pulnix TMC9701) with a wide angles lens (4.9mm / FOV 82^o). The camera output was recoded on a Sony Hi8 camcorder. The camera was mounted on a tripod and an effort was made to reduce the amount of rotation, but rotation was 0.32^o. Figure 27d shows the recovered depth map. The 3d Euclidean rendering of the depth map is shown in Figure 28.

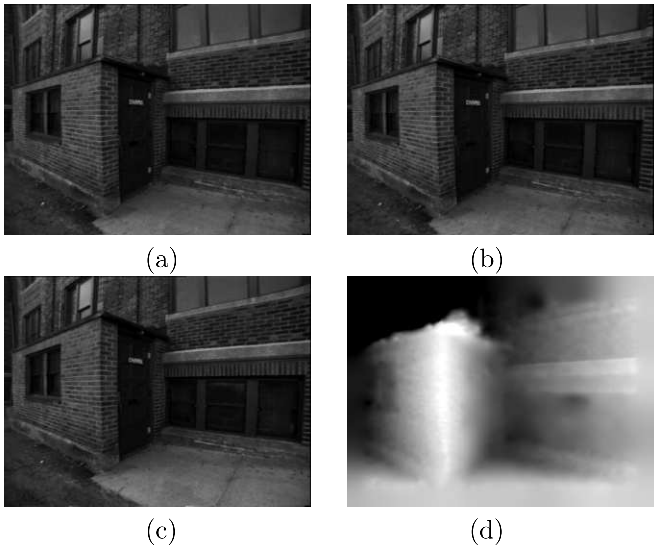


Fig. 27. *Three input views of a side entrance to a building and resulting depth map. Camera was mounted on a tripod but motion included up to* 0.5^o *rotation.*

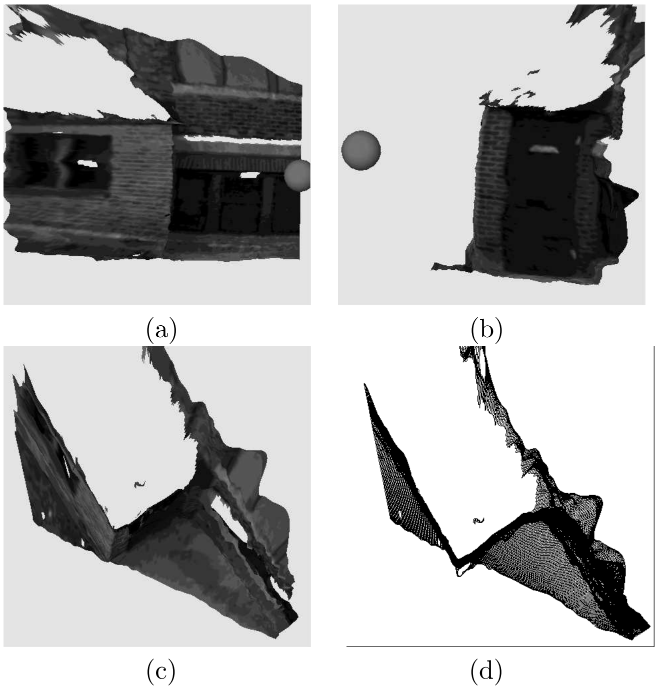


Fig. 28. *3D renderings of the depth map in Figure 27d. The spheres in (a) and (b) indicate the estimated camera location. Note in overhead view (d) true Euclidean structure is recovered with correct* 90^o *angles.*

6. Summary and Future Work

We have presented a general relationship between the spatio-temporal derivatives of three frames and the ego-motion parameters of the two motions. This relationship was derived first for the general case where the ego-motion model comprises of the 27 coefficients of the trilinear tensor, and then for the small-motion model of Longuet-Higgins & Prazdny.

On practical grounds, these relationships are model-based brightness constraints which provide a linear constraint per pixel in the image — thereby providing direct structure and motion estimation that cuts through the aperture problem and without prior detection of feature points. The linear constraints of the 27-parameter projective model are degenerate and cannot lead directly to a unique solution for the tensor coefficients although a unique solution can be found using quadratic admissibility constraints. This added complexity led us to implement in practice the simpler small-motion model of Longuet-Higgins & Prazdny.

The implementation details of these model-based brightness constraints are important and include four critical elements: (i) embedding of the computations within a coarse-to-fine (Gaussian pyramid) framework, (ii) Newton iterations over the brightness constraint equation, (iii) post-processing of smoothness and surface interpolation for obtaining visually pleasing results, and (iv) obtaining both stability and accuracy by assuming a very short focal length for the motion estimation, but using the correct focal length for the image warping.

The algorithm was tested on a set of challenging real image situations which contain very few "good features" (local regions with significant variability of gradient direction) which are necessary for optic flow and discrete point matching algorithms. We obtained a faithful dense depth map of the scene due to the fact that the algorithm is not hindered by the presence of aperture effects.

6.1. Future Work - Collinear Motion

The method, as described so far, fails when the two motions are in the same (or opposite) directions. This is a drawback in many applications (e.g. 3D reconstruction from a monocular image sequence). We present here some early research on ways to overcome this problem. The initial results look promising ([44]). We will investigate the pure translation case because if it fails for pure translation, it will also fail if there is some rotation.

For pure translation:

$$I''_t s^\top t' - I'_t s^\top t'' = 0 \tag{59}$$

and

$$ks^\top t'' + I''_t = 0 \tag{60}$$

$$ks^\top t' + I'_t = 0 \tag{61}$$

If the second translation vector t'' is proportional to the translation vector t' then equation (60) is simply a scaled version of equation (61) adding no new information and the solution is ill conditioned. Another way of interpreting equations (61) and (60) is that a motion t' in one direction will create a change in the image I'_t which exactly the opposite of the change I''_t induced by a motion t'' of equal magnitude and opposite direction to t'.

The LH&P model which assumes ($\frac{t_z}{Z} << 1$) and was used to derive equations (61) and (60) is of course not exact unless the motion in the Z direction is zero, but in any case the contribution to the optic flow due to translation in the Z direction is small unless the field of view is very wide. Therefore, for small motion, the LH&P model will be quite accurate and the equations will be ill-conditioned.

This situation is closely related to the case of reconstruction from line correspondences. There is a critical combination of a line configuration and camera motions which can be stated in the following theorem:

Theorem 2 (Critical configuration of lines and motion) *Let $\mathcal{S}$ be a set of lines in 3D which have a common intersecting line L (i.e., $S \wedge L = 0$ for all $S \in \mathcal{S}$). Let s,s',s'' be the projections of the line $S \in \mathcal{S}$ in the three views: Image 1, Image 2 and Image 3. Let o, o' and o'' be the corresponding camera centers. If o, o' and o'' all lie on a line $S_i \in \mathcal{S}$, then the corresponding lines s,s',s'' do not provide a unique solution to the camera motion.*

We currently have no proof, but simulation experiments confirm the theorem. This line configuration is the Linear Line Complex (LLC) which is examined in [46]. In [46] it is shown that for general motion, the tensor has a linear degeneracy in the case of an LLC, but a unique solution is obtained by taking into account nonlinear constraints. Here, for the specific case where the camera motion is along a line belonging to the LLC, there is a whole family of solutions. The epipole could lie anywhere on the projection of the line L onto the image. Thus, there is one degree of uncertainty in addition to the scale factor ambiguity.

If we take the common line to be the line at infinity on the XY plane we get the particular case where all the lines lie on planes which are parallel to the image plane, and the translations are collinear and also parallel to the image plane. This is a critical condition according to Theorem 2 and the direction of translation in the XY plane cannot be recovered. In such a configuration, lines in Image 1 will be parallel to corresponding lines in Image 2 and Image 3 (assuming no rotation).

How does this relate to our case? By using the first order approximation of the optic flow constraint equation to get the point-line-line correspondences, we have created, as an artifact, a situation where all corresponding lines are parallel. In reality this is not the case, unless, all the lines came from planes parallel to the image plane.

6.1.1. *A Possible Solution*

The Optic Flow Constraint Equation [19] was based on the first order Taylor expansion of $I(x, y, t) = I(x', y', t + \delta t)$ where t is used to denote time (not translation). Keeping the second order terms of the Taylor expansion results in:

$$\begin{aligned} I(x+\delta x, y+\delta y, t+\delta t) = \\ & I(x,y,t) + \delta x I_x + \delta y I_y + \delta t I_t + \\ & \frac{1}{2!}\left(\delta x^2 E_{xx} + \delta y^2 E_{yy} + \delta t^2 E_{tt} + 2\delta x \delta y E_{xy} + \right. \\ & \left. \quad 2\delta x \delta t E_{xt} + 2\delta y \delta t E_{yt}\right) + e \end{aligned} \tag{62}$$

Replacing $\delta x = u'\delta t$ and $\delta y = v'\delta t$ yields:

$$\begin{aligned} I(x+u\delta t, y+v\delta t, t+\delta t) = \\ & I(x,y,t) + u'\delta t I_x + v'\delta t I_y + \delta t I_t + \\ & \frac{1}{2!}\left((u\delta t)^2 E_{xx} + (v'\delta t)^2 E_{yy} + \delta t^2 E_{tt} + \right. \\ & \left. \quad 2u'v'\delta t^2 E_{xy} + 2u'\delta t^2 E_{xt} + 2v'\delta t^2 E_{yt}\right) \\ & + e \end{aligned} \tag{63}$$

We notice that two of the second order terms are also linear with u and v. If we keep those two terms, (at this point, apart from convenience, we have no formal justification why we should keep those two terms and not all the 2nd order terms) and apply the constant brightness assumption:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

we get for the first motion:

$$u'(I_x + I_{xt}\delta t) + v'(I_y + I_{yt}\delta t) + I'_t = 0 \tag{64}$$

Since $I'_x = I_x + I_{xt}\delta t$ and $I'_y = I_y + I_{yt}\delta t$ this becomes:

$$u'I'_x + v'I'_y + I'_t = 0 \tag{65}$$

This is similar to the original work by Horn & Schunk who used:

$$u'\frac{(I'_x + I_x)}{2} + v'\frac{(I'_y + I_y)}{2} + I'_t = 0.$$

Note that I'_x and I'_y are computed at coordinates (x, y) not (x', y'). Similarly for the second motion:

$$u''I''_x + v''I''_y + I''_t = 0 \tag{66}$$

Likewise I''_x, I''_y are the brightness gradients in Image 3 at coordinates (x, y). That leads us to the equation:

$$I''_t s'^{\top} t' - I'_t s''^{\top} t'' = 0 \tag{67}$$

where:

$$s' = \begin{pmatrix} I'_x \\ I'_y \\ -xI'_x - yI'_y \end{pmatrix} \qquad d'' = \begin{pmatrix} I''_x \\ I''_y \\ -xI''_x - yI''_y \end{pmatrix} \tag{68}$$

Since now $s' \neq s''$, the equations are better conditioned. This depends on how much change there was in the images gradients. As we have noted, this depends on whether the surface is parallel to the image plane or not.

A modification is required to the coarse-to-fine framework and the iterative refinement which are described in Section 4.4 and Section4.2 respectively. For motion estimation, Image 1 is warped (using forward warping) towards Image 2 and Image 3. For depth estimation Image 2 and Image 3 are warped towards Image 1.

We have shown an initial solution to the case of collinear motion with pure translation and have successfully implemented these ideas. It is not clear how to transfer this to the general motion case and this remains an open area of study.

References

[1] S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensors. *IEEE Transactions on Visualization and Computer Graphics*, 4(3), 1998. Short version can be found in CVPR'97.

[2] A. Azarbayejani and A. P. Pentland. Recursive estimation of motion, structure and focal length. *pami*, 17(6):562–575, June 1995.

[3] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, Santa Margherita Ligure, Italy, June 1992.

[4] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, Santa Margherita Ligure, Italy, June 1992.

[5] J. Bergen and R. Hingorani. Hierarchical motion-based frame rate conversion. Technical report, David Sarnoff Research Center, 1990.

[6] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[7] R. Deriche and G. Giraudon. Accurate corner detection: An analytical study. In *Proceedings of the International Conference on Computer Vision*, pages 66–70, Osaka, Japan, December 1990.

[8] R. Dutta and M. Snyder. Robustness of correspondence based structure from motion. In *Proceedings of the International Conference on Computer Vision*, pages 106–110, Osaka, Japan, December 1990.

[9] O. D. Faugeras. *Three Dimensional Computer Vision: a Geometric Viewpoint*. The MIT Press, Cambridge, MA, 1993.

[10] O. D. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between N images. In *Proceedings of the International Conference on Computer Vision*, pages 951–956, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[11] C. Fermuller and Y. Aloimonos. Global rigidity constraints in image displacement fields. In *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[12] C. Fermuller and Y. Aloimonos. The confounding of translation and rotation in reconstruction from multiple views. In *CVPR*, Puerto Rico, June 1997. IEEE Computer Society Press, IEEE Computer Society Press.

[13] C. Fermuller and Y. Aloimonos. On the geometry of visual correspondence. *International Journal of Computer Vision*, 21(3):223–47, February 1997.

[14] K. Hanna and N. Okamoto. Combining stereo and motion analysis for direct estimation of scene structure. In *Proceedings of the International Conference on Computer Vision*, pages 357–365, Berlin, Germany, May 1993.

[15] R. Hartley. A linear method for reconstruction from lines and points. In *Proceedings of the International Conference on Computer Vision*, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[16] D. J. Heeger and A. Jepson. Simple method for computing 3d motion and depth. In *Proceedings of the International Conference on Computer Vision*, pages 96–100, Osaka, Japan, December 1990.

[17] J. Heel. Direct estimation of structure and motion from multiple frames. AI Memo 1190, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, March 1990.

[18] A. Heyden. Reconstruction from image sequences by means of relative depths. In *Proceedings of the International Conference on Computer Vision*, pages 1058–1063, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

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

[20] B. K. P. Horn and E. J. Weldon, Jr. Direct methods for recovering motion. *International Journal of Computer Vision*, 2:51–76, 1988.

[21] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *CVPR*, pages 454–460, Seattle, Washington, June 1994.

[22] R. Kumar and P. Anandan. Direct recovery of shape from multiple views: A parallax based approach. In *Proceedings of the International Conference on Pattern Recognition*, Jerusalem, Israel, October 1994.

[23] H. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London B*, 208:385–397, 1980.

[24] H. Longuett-Higgins and K. Prazdny. The interpretation of a moving retinal image. In *Proceedings of the Royal Society of London B*, volume 208, pages 385–397. Royal Society of London, 1980.

[25] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings IJCAI*, pages 674–679, Vancouver, 1981.

[26] I. S. McQuirk. An analog vlsi chip for estimating the focus of expansion. AITR 1577, MIT, 1996.

[27] D. Michaels. *Exploiting Continuity-in-Time in Motion Vision*. PhD thesis, Department of Electrical engineering and Computer Science, Massachusetts Institute of Technology, May 1992.

[28] J. J. More, B. S. Garbow, and K. E. Hillstrom. User guide for minpack-1. Technical Report ANL-80-74, Argonne National Laboratory, Argonne, Illinois, August 1980.

[29] H. H. Nagel. Direct estimation of optical flow and of its derivates. In G. A. Orban and H. H. Nagel, editors, *Artificial and Biological Vision Systems*, Basic Research Series, pages 193–224. Springer-Verlag, 1992.

[30] S. Negahdaripour and B. K. P. Horn. Direct passive navigation. *Pattern Analysis and Machine Intelligence*, 9(1):168–176, 1987.

[31] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition edition, 1992.

[32] J. Reiger and D. T. Lawton. Processing differential image motion. *Journal of the Optical Society of America*, 2:354–359, 1985.

[33] H. Sawhney. 3D geometry from planar parallax. In *CVPR*, pages 929–934, Seattle, Washington, June 1994.

[34] A. Shashua. Algebraic functions for recognition. *PAMI*, 17(8):779–789, 1995.

[35] A. Shashua. Trilinear tensor: The fundamental construct of multiple-view geometry and its applications. In G. Sommer and J. Koenderink, editors, *Algebraic Frames For The Perception Action Cycle*, number 1315 in Lecture Notes in Computer Science. Springer, 1997. Proceedings of the workshop held in Kiel, Germany, Sep. 1997.

[36] A. Shashua and S. Avidan. The rank4 constraint in multiple view geometry. In *Proceedings of the European Conference on Computer Vision*, Cambridge, UK, April 1996.

[37] A. Shashua and N. Navab. Relative affine structure: Canonical model for 3D from 2D geometry and applications. *PAMI*, 18(9):873–883, 1996.

[38] A. Shashua and P. Anandan. Trilinear constraints revisited: Generalized trilinear constraints and the tensor brightness constraint. In *Proceedings of the ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.

[39] A. Shashua and K. J. Hanna. The tensor brightness constraints: Direct estimation of motion revisited. Technical report, Technion, Haifa, Israel, November 1995.

[40] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *Proceedings of the International Conference on Computer Vision*, pages 920–925, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[41] S. Soatto and P. Perona. Motion from fixation. In *CVPR*, pages 817–824, San Fransico, CA, June 1996.

[42] M. Spetsakis and J. Aloimonos. A unified theory of structure from motion. In *Proceedings of the ARPA Image Understanding Workshop*, 1990.

[43] G. P. Stein. Lens distortion calibration using point correspondences. In *CVPR*, Puerto Rico, June 1997.

[44] G. P. Stein. *Geometric and Photometric Constraints: Motion and Structure from Three Views*. PhD thesis, M.I.T Artificial Intelligence Laboratory, February 1998.

[45] G. P. Stein and A. Shashua. Model based brightness con-

straints: On direct estimation of structure and motion. In *CVPR*, Puerto Rico, June 1997.

[46] G. P. Stein and A. Shashua. On degeneracy of linear reconstruction from three views: Linear line complex and applications. In *Proceedings of the European Conference on Computer Vision*, June 1998.

[47] R. Szeliski and S. B. Kang. Direct methods for visual scene reconstruction. In *Proceedings IEEE Workshop on Representation of Visual Scenes*, Cambridge, MA, June 1995.

[48] T. Y. Tian, C. Tomasi, and D. J. Heeger. Comparison of approaches to egomotion computation. In *CVPR*, pages 315–320, San Francisco, California, June 1996.

[49] C. Tomasi and T. Kanade. Shape and motion from image streams – a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[50] B. Triggs. Matching constraints and the joint image. In *Proceedings of the International Conference on Computer Vision*, pages 338–343, Cambridge, MA, June 1995. IEEE Computer Society Press, IEEE Computer Society Press.

[51] J. Weng, N. Ahuja, and T. Huang. Two view matching. In *CVPR*, pages 64–73, Tampa, Florida, December 1989.

[52] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, California, 1990.