Recovery of Ego-Motion Using Region Alignment

Michal Irani, *Member, IEEE*,
Benny Rousso, *Student Member, IEEE*,
and Shmuel Peleg, *Member, IEEE*

Abstract—A method for computing the 3D camera motion (the *ego-motion*) in a static scene is described, where initially a detected 2D motion between two frames is used to align corresponding image regions. We prove that such a 2D registration removes all effects of camera rotation, even for those image regions that remain misaligned. The resulting *residual* parallax displacement field between the two region-aligned images is an *epipolar field* centered at the FOE (Focus-of-Expansion). The 3D camera translation is recovered from the epipolar field. The 3D camera rotation is recovered from the computed 3D translation and the detected 2D motion. The decomposition of image motion into a *2D parametric motion* and *residual epipolar parallax displacements* avoids many of the inherent ambiguities and instabilities associated with decomposing the image motion into its *rotational* and *translational* components, and hence makes the computation of ego-motion or 3D structure estimation more robust.

Index Terms—Motion analysis, ego motion, video stabilization, plane-plus-parallax.

1 Introduction

The motion observed in an image sequence can be caused by camera motion (ego-motion) and by motions of objects moving in the scene. In this paper we address the case of a camera moving in a static scene. Complete 3D motion estimation is difficult since the image motion at every pixel depends, in addition to the six parameters of the camera motion, on the depth at the corresponding scene point. To overcome this difficulty, additional constraints are usually added to the motion model or to the scene structure.

3D motion is often estimated from the optical or normal flow derived between two frames [1], [11], [23], or from the correspondence of distinguished features (points, lines, contours) extracted from successive frames [24], [12], [8]. Both approaches depend on the accuracy of the feature detection, which can not always be assured. Methods for computing the ego-motion directly from image intensities were also suggested [10], [13].

Camera rotations and translations can induce similar image motions [2], [9] causing ambiguities in their interpretation. The problem of recovering the 3D camera motion from a flow field is therefore an ill-conditioned problem, since small errors in the 2D flow field usually result in large perturbations in the 3D motion [2]. At depth discontinuities, however, it is much easier to distinguish between the effects of camera rotations and camera translations, as the image motion of neighboring pixels at different depths will have similar rotational components, but different translational components [20].

• M. Irani is with the Department of Applied Math and Computer Science, The Weizmann Institute of Science, 76100 Rehovot, Israel.
E-mail: irani@wisdom.weizmann.ac.il.
• B. Rousso and S. Peleg are with the Institute of Computer Science, The Hebrew University of Jerusalem, 91904 Jerusalem, Israel.
E-mail: {tokyo, peleg}@cs.huji.ac.il.

Manuscript received June 1993; revised Dec. 9, 1996. Recommended for acceptance by J. Aloimonos.
For information on obtaining reprints of this article, please send e-mail to: transpami@computer.org, and reference IEEECS Log Number P96131.

In this paper a method is introduced for computing the ego-motion based on a decomposition of the image motion into a 2D parametric transformation and a residual parallax displacement field. This decomposition can be obtained more robustly, and avoids many of the inherent ambiguities and instabilities associated with decomposing a flow field into its rotational and translational components.

Initially, methods to compute the 2D motion of one image region [15], [16], [5], [4] are used to detect an image region and compute its 2D parametric motion between two image frames. The two frames are then registered according to the computed 2D parametric transformation. This step removes all effects of the camera rotation, even for the misaligned image regions. The residual parallax displacement field between the 2D region-aligned images is an epipolar field centered at the FOE, which can be computed from the epipolar field. When calibration information is provided, the 3D camera translation is recovered. The 3D rotation is estimated by solving a small set of linear equations, which depend on the computed 3D translation and the detected 2D parametric motion.

As opposed to other methods which use motion parallax for 3D estimation [20], [21], [19], [8], our method does not rely on parallax information at depth discontinuities (where flow computation is likely to be inaccurate). The residual displacements after 2D alignment provide a dense and more reliable parallax field.

The advantage of the proposed technique is in its simplicity and in its robustness. No prior detection and matching are assumed, it requires solving only small sets of linear equations, and each computational step is stated as an overdetermined highly constrained problem which is numerically stable.

Similar approaches are described in [17], [25], [18], [26], [14] and are often referred to by the name "plane-plus-parallax," since the estimated 2D parametric transformation frequently corresponds to the induced homography of a 3D planar surface in the scene.

2 Ego-Motion From 2D Image Motion

2.1 Basic Model and Notations

Let (X, Y, Z) denote the Cartesian coordinates of a scene point with respect to the camera (see Fig. 1), and let (x, y) denote the corresponding coordinates in the image plane. The image plane is located at the focal length: $Z = f_c$. The perspective projection of a scene point $P = (X, Y, Z)^t$ on the image plane at a point $p = (x, y)^t$ is expressed by:

$$p = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{X}{Z} & f_c \\ \frac{Y}{Z} & f_c \end{bmatrix} \tag{1}$$

The camera motion has two components: a translation $T = (T_X, T_Y, T_Z)^t$ and a rotation $\Omega = (\Omega_X, \Omega_Y, \Omega_Z)^t$. Due to the camera motion the scene point $P = (X, Y, Z)^t$ appears to be moving relative to the camera with rotation $-\Omega$ and translation $-T$, and is therefore observed at new world coordinates $P' = \left(X', Y', Z'\right)^t$, expressed by:

$$P' = M_{-\Omega} \cdot P - T \tag{2}$$

where $M_{-\Omega}$ is the matrix corresponding to a rotation by $-\Omega$.

With a small field of view and a relatively small camera rotation [1], the 2D displacement (u, v) of an image point (x, y) in the image plane can be expressed by [22], [1]:

$$\begin{bmatrix}u\\v\end{bmatrix} = \begin{bmatrix} -f_c\left(\frac{T_X}{Z}+\Omega_Y\right)+x\frac{T_Z}{Z}+y\Omega_Z - x^2\frac{\Omega_Y}{f_c}+xy\frac{\Omega_X}{f_c} \\ -f_c\left(\frac{T_Y}{Z}-\Omega_X\right)-x\Omega_Z+y\frac{T_Z}{Z}-xy\frac{\Omega_Y}{f_c}+y^2\frac{\Omega_X}{f_c}\end{bmatrix} \quad (3)$$

All points (X, Y, Z) of a planar surface in the 3D scene satisfy a plane equation $Z = A + B \cdot X + C \cdot Y$, which can be expressed in terms of image coordinates by using (1) as:

$$\frac{1}{Z} = \alpha + \beta \cdot x + \gamma \cdot y \quad (4)$$

where $\alpha = \frac{1}{A}$, $\beta = -\frac{B}{f_c A}$, and $\gamma = -\frac{C}{f_c A}$. In a similar manipulation to that in [1], substituting (4) in (3) yields the *2D quadratic transformation*:

$$\begin{bmatrix}u\\v\end{bmatrix} = \begin{bmatrix} a + b\cdot x + c \cdot y + g\cdot x^2 + h \cdot xy \\ d + e\cdot x + f\cdot y + g\cdot xy + h\cdot y^2 \end{bmatrix} \quad (5)$$

where:

$$\begin{aligned} a &= -f_c\alpha T_X - f_c\Omega_Y & e &= -\Omega_Z - f_c\beta T_Y \\ b &= \alpha T_Z - f_c\beta T_X & f &= \alpha T_Z - f_c\gamma T_Y \\ c &= \Omega_Z - f_c\gamma T_X & g &= -\frac{\Omega_Y}{f_c} + \beta T_Z \\ d &= -f_c\alpha T_Y + f_c\Omega_X & h &= \frac{\Omega_X}{f_c} + \gamma T_Z \end{aligned} \quad (6)$$

Equation (5), expressed by eight parameters (*a, b, c, d, e, f, g, h*), describes the 2D parametric image motion of a 3D planar surface. The quadratic transformation in (5) is a good approximation to the 2D projective transformation assuming a small field of view and a small rotation.

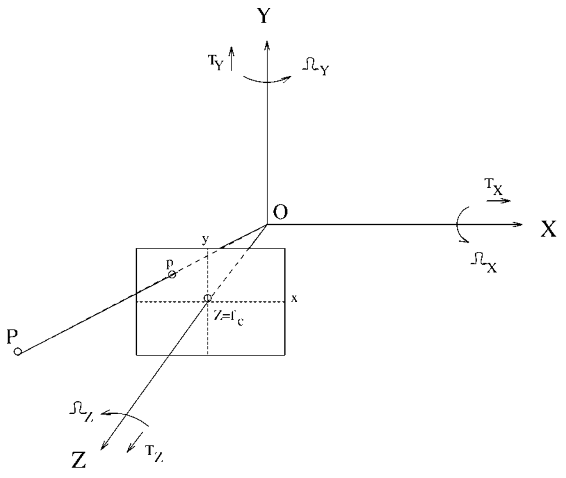

Fig. 1. The coordinate system. The coordinate system (X, Y, Z) is attached to the camera, and the corresponding image coordinates (x, y) on the image plane are located at $Z = f_c$. A point $P = (X, Y, Z)^t$ in the world is projected onto an image point $p = (x, y)^t$. $T = (T_X, T_Y, T_Z)^t$ and $\Omega = (\Omega_X, \Omega_Y, \Omega_Z)^t$ represent the relative translation and rotation of the camera in the scene.

2.2 General Framework of the Algorithm

In this section we present a scheme which utilizes the robustness of the 2D motion computation for computing 3D motion between two consecutive frames:

1) A single image region with a 2D *parametric* image motion is automatically detected (Section 3). As mentioned in Section 2.1, this image region typically corresponds to a planar surface in the scene, or to a distant part of the scene.
2) The two frames are registered according to the computed 2D parametric motion of a detected image region. This alignment of an image region cancels the rotational component of the camera motion for the entire scene (proved in Section 2.3).
3) The FOE (and the camera translation) is computed from the residual epipolar displacement field between the two registered frames (Section 2.4).
4) The 3D rotation of the camera is computed (Section 2.5) from the 2D motion parameters of the detected image region and the 3D translation.

2.3 Canceling Camera Rotation by 2D Region Alignment

At this stage we assume that a single image region with a parametric 2D image motion has been detected, and that the 2D image motion of that region has been computed (see Section 3).

Let $(u(x, y), v(x, y))$ denote the 2D image motion of the entire scene from frame f_1 to frame f_2, and let $(u_s(x, y), v_s(x, y))$ denote the 2D image motion of a single image region (the detected image region) between the two frames. Let S denote the 3D surface corresponding to the detected image region, with depths $Z_s(x, y)$. As mentioned in Section 2.1, (u_s, v_s) can be expressed by a 2D parametric transformation (5) when S satisfies one of the following conditions:

1) S is a *planar* surface in the 3D scene,
2) S is an *arbitrary* 3D scene, but the camera's motion is only rotation or zoom, or
3) S is a portion of the scene that is distant enough from the camera compared to the camera translation.

Assuming the existence of such a surface S in the scene is not a severe restriction, as most indoor scenes contain a planar surface, and in outdoor scenes the ground or any distant object can serve as such a surface. Note also that only the 2D motion parameters $(u_s(x, y), v_s(x, y))$ of the 3D surface S are estimated. Neither the 3D structure of S nor the 3D motion parameters are estimated at this point.

Let f_1^R denote the frame obtained by warping the entire frame f_1 towards frame f_2 according to the 2D parametric transformation (u_s, v_s) extended to the entire frame. This warping will bring the image region R, corresponding to the detected surface S, into perfect alignment between f_1^R and f_2. In the warping process, each pixel (x, y) in f_1 is displaced by $(u_s(x, y), v_s(x, y))$ to form f_1^R. Points that are not located on the parametric surface S (i.e., $Z(x, y) \neq Z_s(x, y)$) will *not* be in registration between f_1^R and f_2. We will now show that the residual 2D image displacements between the two registered frames (f_1^R and f_2) forms an epipolar field centered at the original FOE, i.e., affected only by the camera translation T.

Let $P_1 = (X_1, Y_1, Z_1)^t$ denote the 3D scene point projected onto $p_1 = (x_1, y_1)^t$ in f_1. According to (1):

$$P_1 = \left(x_1 \frac{Z_1}{f_c}, y_1 \frac{Z_1}{f_c}, Z_1 \right)^t.$$

Due to the camera motion (Ω, T) from frame f_1 to frame f_2, the point P_1 will be observed in frame f_2 at $p_2 = (x_2, y_2)^t$, which corresponds to the 3D scene point $P_2 = (X_2, Y_2, Z_2)^t$. According to (2):

$$P_2 = M_{-\Omega} \cdot P_1 - T. \tag{7}$$

The warping of f_1 by (u_s, v_s) to form f_1^R is equivalent to applying the camera motion (Ω, T) to the 3D points as if they are all located on the surface S (i.e., with depths $Z_s(x, y)$). Let P_s denote the 3D point on the surface S which corresponds to the pixel (x, y) with depth $Z_s(x, y)$. Then:

$$P_s = \begin{bmatrix} x_1 \frac{Z_s}{f_c} \\ y_1 \frac{Z_s}{f_c} \\ Z_s \end{bmatrix} = \frac{Z_s}{Z_1} \cdot \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \frac{Z_s}{Z_1} \cdot P_1 \tag{8}$$

After the image warping, P_s is observed in f_1^R at $p^R = (x^R, y^R)^t$, which corresponds to a 3D scene point P^R. Therefore, according to (2) and (8):

$$P^R = M_{-\Omega} \cdot P_s - T = \frac{Z_s}{Z_1} \cdot M_{-\Omega} \cdot P_s - T$$

and therefore:

$$P_1 = \frac{Z_1}{Z_s} \cdot M_{-\Omega}^{-1} \cdot \left(P^R + T \right) \tag{9}$$

By substituting (9) in (7), P^R can be expressed as:

$$P^R = \frac{Z_s}{Z_1} \cdot P_2 + \left(1 - \frac{Z_s}{Z_1} \right) \cdot (-T) \tag{10}$$

Equation (10) shows that P^R is independent of the camera rotation Ω. Moreover, P^R is on the straight line passing through P_2 and $-T$. Therefore, the projection of P^R on the image plane (p^R) is on the straight line passing through the projection of P_2 (i.e., p_2) and the projection of $-T$ (i.e., the FOE). This means that p^R is found on the radial line emerging from the FOE toward p_2. In other words, the residual image displacements between the registered frames f_1^R and f_2 (i.e., $p^R - p_2$) form an epipolar field centered at the FOE. (Note that the magnitudes of the residual displacements depend on the scene structure, $\frac{Z_s}{Z_1}$, however their directions do not.)

In Fig. 2, the optical flow is displayed before and after registration of two frames according to the computed 2D motion parameters of the image region (the region in this case is the wall at the back of the scene). The optical flow is given for display purposes only, and was not used for registration. After registration, the rotational component of the optical flow was canceled for the entire scene, and almost all flow vectors point towards the real FOE (Fig. 2c). Before registration (Fig. 2b) the FOE mistakenly appears to be located elsewhere (in the middle of the frame). This is due to the ambiguity caused by the rotation around the Y-axis, which visually appears as a translation along the X-axis. This ambiguity is resolved by the 2D registration.

2.4 Computing Camera Translation

Once the rotation is canceled by the 2D alignment of the detected image region, the ambiguity between image motion induced by 3D rotation and that induced by 3D translation no longer exists (see Section 2.3). Having cancelled effects of camera rotation, the residual displacement field is directed towards, or away from, the FOE. The computation of the FOE therefore becomes overdetermined and numerically stable, as there are only two unknowns to the problem: the 2D coordinates of the center of the epipolar field (i.e, FOE) in the image plane.

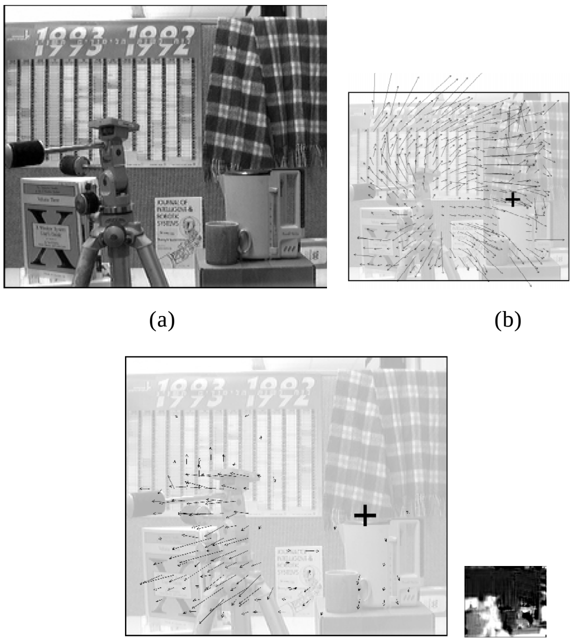

Fig. 2. The optical flow before and after 2D alignment of the image region corresponding to the the wall. The camera was translating and rotating, and the real FOE is marked by {**+**}. The optical flow is given only for display purposes, and it is *not* used for the registration. (a) One of the frames in the sequence. (b) The optical flow to another frame (before registration), overlayed on Fig. 2a. The FOE mistakenly appears to be in the wrong location (in the middle of the frame). This is due to the ambiguity caused by the camera rotation around the Y-axis. (c) The optical flow after 2D alignment of the wall. The flow is induced by pure camera translation (after the camera rotation was canceled), and points to the correct FOE. (d) An example of depth map, computed using the recovered ego-motion. Bright regions correspond to near objects.

To locate the FOE, the parallax optical flow between the registered frames is computed, and the FOE is located using a search method similar to that described in [20]. Candidates for the FOE are sampled over a half sphere and projected onto the image plane. For each such candidate, a global error measure is computed from local deviations of the flow field from the radial lines emerging from the candidate FOE. The search process is repeated by refining the sampling (on the sphere) around good FOE candidates. After a few refinement iterations, the FOE is taken to be the candidate with the smallest error.

Since the problem of locating the FOE in a purely translational (epipolar) flow field is a highly overdetermined problem, the computed flow field need not be accurate. This is opposed to most methods which try to compute the ego-motion from the flow field, and require an accurate flow field in order to resolve the rotation-translation ambiguity [2]. Furthermore, the residual flow field can be estimated more accurately than general flow, as it is globally constrained to lie on an epipolar field. Once the FOE is estimated, and given camera calibration information, the 3D camera translation (T_X, T_Y, T_Z) is recovered.

2.5 Computing Camera Rotation

Let (a, b, c, d, e, f, g, h) be the 2D motion parameters of an image region as expressed by (5). Given these 2D motion parameters and the 3D translation parameters of the camera (T_X, T_Y, T_Z), the 3D rotation parameters of the camera $(\Omega_X, \Omega_Y, \Omega_Z)$ (as well as the surface parameters of the plane (α, β, γ)) can be obtained by solving (6), which is a set of *eight* linear equations in *six unknowns*.

Since the parameters g and h in the quadratic transformation of (5) are second order terms, they are not as reliable as the other six parameters (a, b, c, d, e, f). Therefore, whenever possible (when the set of (6) is numerically overdetermined), we avoid using the last two equations (of g and h), and use only the first six. This yields more accurate results.

2.6 Experimental Results

The camera motion between the two frames in Fig. 2 was: $(T_X, T_Y, T_Z) = (1.7_{cm}, 0.4_{cm}, 12_{cm})$ and $(\Omega_X, \Omega_Y, \Omega_Z) = (0^\circ, -1.8^\circ, -3^\circ)$. The computation of the 3D motion parameters of the camera (after calibrating T_Z to 12_{cm}, as $\vec{T}$ can only be determined up to a scale factor, yielded: $(T_X, T_Y, T_Z) = (1.68_{cm}, 0.16_{cm}, 12_{cm})$ and $(\Omega_X, \Omega_Y, \Omega_Z) = (-0.05^\circ, -1.7^\circ, -3.25^\circ)$.

Once the 3D motion parameters of the camera are computed, the 3D scene structure can be reconstructed using a scheme similar to that suggested in [10]. Correspondences between small image patches (currently 5×5 pixels) are computed only along the radial lines emerging from the FOE (taking the rotations into account). The depth map is computed from the magnitude of these displacements. In Fig. 2d, the computed inverse depth map of the scene $\left(\frac{1}{Z(x,y)}\right)$ is displayed. Similar approaches to 3D shape recovery have since been suggested by [25], [18], [26], [14].

Fig. 3 shows an example where the ego-motion estimation was used to electronically stabilize (i.e., remove camera jitter) a sequence obtained by a hand held camera.

3 Computing a 2D Parametric Motion

We use the method described in [16] to detect a 2D parametric transformation of an image region. This method is briefly described in this section. Other methods for computing a 2D parametric region motion [7], [3] can be used as well.

Let R be an image region that has a single 2D parametric transformation $\mathbf{q}$ between two frames, $I(x, y, t)$ and $I(x, y, t + 1)$. $\mathbf{q}$ is a quadratic transformation expressed by eight parameters $\mathbf{q} = (a, b, c, d, e, f, g, h)$ (see (5)). $(u, v) = (u(x, y; \mathbf{q})), v(x, y; \mathbf{q}))$ is the 2D motion field described by $\mathbf{q}$. To solve for the unknown parameters of $\mathbf{q}$, the following SSD error measure is minimized:

$$\begin{aligned} E^{(t)}(\mathbf{q}) &= \sum_{(x,y)\in R} \Big(I(x,y,t) - I(x+u, y+v, t+1)\Big)^2 \\ &\approx \sum_{(x,y)\in R} \Big(uI_x + vI_y + I_t\Big)^2 \end{aligned} \tag{11}$$

The objective function E is minimized via the Gauss-Newton optimization technique, over a coarse-to-fine multi-resolution data structure. Let $\mathbf{q}_i$ denote the current estimate of the quadratic parameters. After warping the inspection image $(I(x, y, t + 1))$ towards the reference image $(I(x, y, t))$ using the parametric transformation $\mathbf{q}_i$, an incremental estimate $\boldsymbol{\delta}\mathbf{q}$ can be determined. After iterating certain number of times within a pyramid level, the process continues at the next finer level [6], [5], [16].

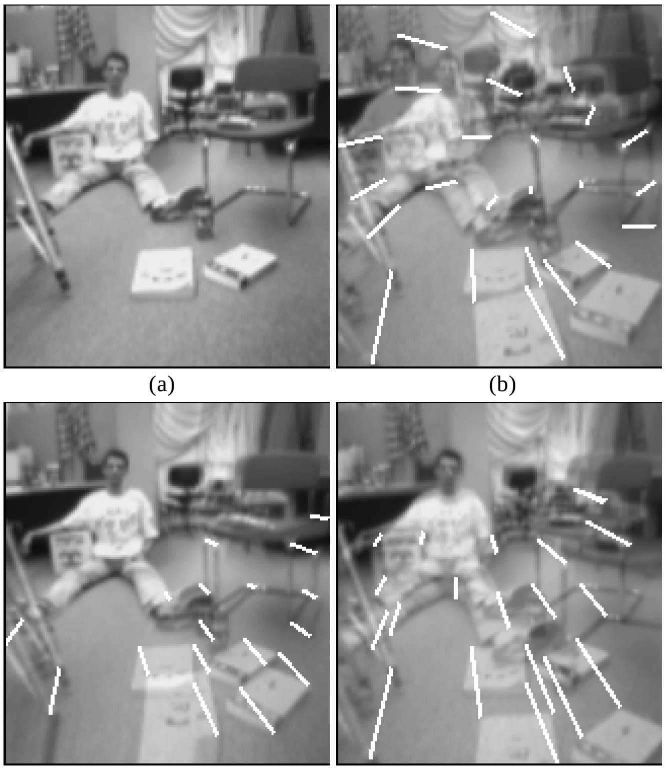
(a) (b) (c) (d)

Fig. 3. Camera stabilization. (a) One of the frames in the sequence. (b) The average of two frames, having both rotation and translation. The white lines display the image motion. (c) The average of the two frames after (automatic) 2D alignment of the shirt. Only effects of camera translation remain. (d) The average of the two frames after recovering the ego motion, and canceling the camera rotation. This results in a 3D-stabilized pair of images (i.e., no camera jitter).

When the above technique is applied to a region R, the reference and the inspection images are registered so that the image region R is aligned. However, a region of support R of an image segment with a single 2D parametric motion is not known a priori. To allow for automatic detection and locking onto a single 2D parametric image motion, a robust version of this scheme is applied [16], [4]. The robust version of the algorithm incorporates two additional mechanisms to the above described scheme:

1) *Outlier Rejection:* The local misalignments at each iteration provide weights for the weighted-least-squares regression process of the next iteration.
2) *Progressive Model Complexity:* The complexity of the 2D parametric motion model used in the regression process is gradually increased with the progression of the iterative process and the outlier rejection. Initially a simple 2D uniform displacement (two parameters) is used, and is gradually refined to a 2D affine transformation (six parameters) and further to a 2D quadratic transformation (eight parameters). The progressive complexity scheme focuses first on the most stable constant terms (a and d), then further refines them along with the linear terms (b, c, e, f), and finally refines all parameters along with the least stable quadratic terms (h and g). This provides the algorithm with increased stability and locking capabilities, thus avoids converging

into local minima.

The increased robustness and accuracy of the global 2D parametric motion estimation in comparison to flow-based estimation methods is due to the following: Optical flow estimation suffers from inaccuracies due to lack of local texture (i.e., within small windows). These inaccuracies lead to large errors in the interpretation the image motion in terms of its rotational and translational components [2]. 2D parametric estimation, on the other hand, is expressed in terms of few parameters (e.g., eight), yet has a substantially larger region of support in the image. Therefore, the "flow" estimation of a 2D parametric motion is highly constrained and well conditioned. Decomposing the image motion into a 2D parametric transformation and a residual (epipolar) parallax displacement field benefits from this property.

4 Concluding Remarks

A method for computing ego-motion in static scenes was introduced. At first, an image region with a dominant 2D parametric transformation is detected, and its 2D motion parameters between successive frames are computed. The 2D motion is then used for image warping, which cancels the rotational component of the 3D camera motion for the *entire* image, and reduces the problem to a pure 3D translation case. The FOE and the 3D camera translation are computed from the 2D registered frames, and the 3D rotation is computed using a small set of linear equations.

The advantage of the presented technique is in its simplicity, and in the robustness and stability of each computational step. The interpretation of the image motion in terms of a 2D parametric transformation and a residual (epipolar) parallax displacement field, can be obtained more robustly, and avoids many of the inherent ambiguities and instabilities associated with decomposing a flow field into its rotational and translational components. Hence, the proposed method provides increased numerical stability and computational efficiency. There are no severe restrictions on the camera motion or on the 3D structure of the environment. Most steps use only image intensities, and the optical flow is used only for extracting the FOE in the case of pure epipolar field, which is an overdetermined problem and hence does not require accurate optical flow. The inherent problems associated with optical flow or with feature matching are therefore avoided.

Acknowledgments

This research has been sponsored by ARPA through the U.S. Office of Naval Research under Grant N00014-93-1-1202, R&T Project Code 4424341-01.

References

[1] G. Adiv, "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 7, no. 4, pp. 384–401, July 1985.

[2] G. Adiv, "Inherent Ambiguities in Recovering 3D Motion and Structure from a Noisy Flow Field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 477–489, May 1989.

[3] S. Ayer and H. Sawhney, "Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and mdl Encoding," *Int'l Conf. Computer Vision*, pp. 777–784, Cambridge, Mass., June 1995.

[4] M. Ben-Ezra, S. Peleg, and B. Rousso, "Motion Segmentation Using Convergence Properties," *Proc. ARPA IU Workshop*, Nov. 1994.

[5] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation," *European Conf. Computer Vision*, pp. 237–252, Santa Margarita Ligure, Italy, May 1992.

[6] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg, "A Three-Frame Algorithm for Estimating Two-Component Image Motion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 886–895, Sept. 1992.

[7] M.J. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields," *CVGIP: Image Understanding*, vol. 63, no. 1, pp. 75–104, Jan. 1996.

[8] R. Chipolla, Y. Okamoto, and Y. Kuno, "Robust Structure from Motion Using Motion Parallax," *Int'l Conf. Computer Vision*, pp. 374–382, Berlin, May 1993.

[9] K. Daniilidis and H.-H. Nagel, "The Coupling of Rotation and Translation in Motion Estimation of Planar Surfaces," *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 188–193, June 1993.

[10] K. Hanna, "Direct Multi-Resolution Estimation of Ego-Motion and Structure from Motion," *IEEE Workshop Visual Motion*, pp. 156–162, Princeton, N.J., Oct. 1991.

[11] D.J. Heeger and A. Jepson, "Simple Method for Computing 3d Motion and Depth," *Int'l Conf. Computer Vision*, pp. 96–100, 1990.

[12] B.K.P. Horn, "Relative Orientation," *Int'l J. Computer Vision*, vol. 4, no. 1, pp. 58–78, June 1990.

[13] B.K.P. Horn and E.J. Weldon, "Direct Methods for Recovering Motion," *Int'l J. Computer Vision*, vol. 2, no. 1, pp. 51–76, June 1988.

[14] M. Irani and P. Anandan, "Parallax Geometry of Pairs of Points for 3D Scene Analysis," *European Conf. Computer Vision*, pp. I:17–30, Cambridge, England, Apr. 1996.

[15] M. Irani, B. Rousso, and S. Peleg, "Detecting and Tracking Multiple Moving Objects Using Temporal Integration," *European Conf. Computer Vision*, pp. 282–287, Santa Margarita Ligure, Italy, May 1992.

[16] M. Irani, B. Rousso, and S. Peleg, "Computing Occluding and Transparent Motions," *Int'l J. Computer Vision*, vol. 12, no. 1, pp. 5–16, Jan. 1994.

[17] M. Irani, B. Rousso, and S. Peleg, "Recovery of Ego-Motion Using Image Stabilization," *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 454–460, Seattle, June 1994.

[18] R. Kumar, P. Anandan, and K. Hanna, "Direct Recovery of Shape from Multiple Views: A Parallax Based Approach," *Proc 12th ICPR*, 1994.

[19] J.M. Lawn and R. Cipolla, "Epipolar Estimation Using Affine Motion-Parallax," *BMVC93*, 1993.

[20] D.T. Lawton and J.H. Rieger, "The Use of Difference Fields in Processing Sensor Motion," *ARPA IU Workshop*, pp. 78–83, June 1983.

[21] C.H. Lee, "Structure and Motion from Two Perspective Views via Planar Patch," *Int'l Conf. Computer Vision*, pp. 158–164, 1988.

[22] H.C. Longuet-Higgins, "Visual Ambiguity of a Moving Plane," *Proc. Royal Society of London B*, vol. 223, pp. 165–175, 1984.

[23] S. Negahdaripour and S. Lee, "Motion Recovery from Image Sequences Using First-Order Optical Flow Information," *IEEE Workshop Visual Motion*, pp. 132–139, Princeton, N.J., Oct. 1991.

[24] F. Lustman, O.D. Faugeras, and G. Toscani, "Motion and Structure from Motion from Point and Line Matching," *Proc. First Int'l Conf. Computer Vision*, pp. 25–34, London, 1987.

[25] H. Sawhney, "3d Geometry from Planar Parallax," *IEEE Conf. Computer Vision and Pattern Recognition*, June 1994.

[26] A. Shashua and N. Navab, "Relative Affine Structure: Theory and Application to 3d Reconstruction from Perspective Views," *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 483–489, Seattle, June 1994.