Recovering 3D Motion of Multiple Objects Using Adaptive Hough Transform

Tina Yu Tian and Mubarak Shah, *Member*, *IEEE*

Abstract—We present a method to determine 3D motion and structure of multiple objects from two perspective views, using adaptive Hough transform. In our method, segmentation is determined based on a 3D rigidity constraint. Instead of searching candidate solutions over the entire five-dimensional translation and rotation parameter space, we only examine the two-dimensional translation space. We divide the input image into overlapping patches, and, for each sample of the translation space, we compute the rotation parameters of patches using least-squares fit. Every patch votes for a sample in the five-dimensional parameter space. For a patch containing multiple motions, we use a redescending M-estimator to compute rotation parameters of a dominant motion within the patch. To reduce computational and storage burdens of standard multidimensional Hough transform, we use adaptive Hough transform to iteratively refine the relevant parameter space in a "coarse-to-fine" fashion. Our method can robustly recover 3D motion parameters, reject outliers of the flow estimates, and deal with multiple moving objects present in the scene. Applications of the proposed method to both synthetic and real image sequences are demonstrated with promising results.

Index Terms—Multiple-motion analysis, segmentation, structure-from-motion, robust estimation, adaptive Hough transform.

1 Introduction

MOTION in an image sequence can be produced by a camera moving in an environment and/or several independently moving objects. The interpretation of motion information consists of segmenting multiple moving objects and recovering 3D motion parameters and structure for each object. A lot of effort has been devoted to the egomotion recovery problem, in which scenes of a static environment taken by a moving camera are analyzed. When several independently moving objects are present in the scene, the complete 3D motion estimation is difficult, since each moving object occupies a small field of view. A robust method for motion recovery, i.e., a method insensitive to image motion measurement noise and small field of view, is required to solve this problem.

The most common approach for motion analysis has two phases: computation of the optical flow field and interpretation of this flow field. For multiple-motion analysis, the essential part is segmentation of independently moving objects. One approach to segmentation is to detect motion boundaries by applying edge detectors to optical flow field (e.g., [18]). However, the optical flow at each pixel depends not only on 3D motion parameters but on corresponding depth, thus, segmentation by applying only edge-detection techniques to the flow field cannot distinguish between real motion boundaries and depth discontinuities. Another approach for segmenting multiple moving objects is based on the set of coherent 2D motion parameters, independent of depth values. This approach [2], [4], [12], [22] exploits 2D parametric motion approximations, such as affine transformation and projective transformation. However, affine transformation is not always valid when the moving object is relatively large and close to the camera. Projective transformation [1] is based on the assumption that the entire scene consists of piecewise planar surfaces, which is not always true. These parametric models ignore the higher-order optical flow information, and thus may yield incorrect motion segmentation (e.g., over-segmentation). Moreover, using a 2D motion model to segment a 3D scene might lead to ambiguities.

In this work, we attempt to solve the problem of 3D motion recovery given two perspective views for an arbitrary scene which may contain several moving objects with possible camera motion. The work described here has the following characteristics:

- No assumption about the scene (e.g., piecewise planar surface, known depth, etc.) or type of motion has been made.
- Since 3D segmentation is preferable to 2D segmentation, in general, segmentation in this work is determined based on a 3D rigidity constraint.
- Instead of searching the candidate solutions over the entire five-dimensional translation and rotation parameter space, only the two-dimensional translation space is examined. The input image is divided into overlapping patches, and, for each sample of the translation space, the rotation parameters of patches are computed by using least-squares fit. Every patch votes for a sample in the five-dimensional parameter space.
- For an image patch containing multiple motions, a redescending M-estimator is used to compute rotation parameters of a dominant motion within the patch. The M-estimation problem is solved using an iterative weighted least-squares method.
- To reduce computational and storage burdens of standard multidimensional Hough transform, adaptive Hough transform is applied to iteratively refine the relevant parameter space in a "coarse-to-fine" fashion.
- The method robustly rejects outliers of optical flow estimates by applying a global Hough transform and a robust estimation technique.
- The method can perform the complete 3D motion recovery when multiple moving objects are present in the scene.

In the next section, we review some previous work on 3D motion segmentation. In Section 3, we present a method for motion estimation using the Hough transform. In Section 4, we describe a robust motion estimation method for multiple objects. Next, we present an algorithm using Adaptive Hough transform. In Section 6, we demonstrate the proposed method on both synthetic and real images. Finally, in Section 7, we conclude and discuss limitations.

2 Previous Work

The previous work on 3D motion segmentation is divided into the methods assuming orthographic projection and perspective projection.

2.1 Orthographic Projection

Thompson et al. [17] combined an orthographic structure-from-motion algorithm and a least median squares (LMedS) method to solve for the relative motion of camera and background, rejecting outliers corresponding to moving objects. Since LMedS regression assumes that at least half of the data points are inliers, this scheme cannot be applied to any scene containing more than two moving objects. Tomasi and Kanade [20] proposed that, under orthographic projection, the measurement matrix, **W**, of feature trajectories can be factored into motion matrix and shape matrix, **V**, using singular value decomposition. Under this framework, Gear [7], and Costeira and Kanade [6] developed the segmentation methods. Each column of **W** represents a single feature point tracked

• *The authors are with the Computer Science Department of the University of Central Florida, Orlando, FL 32816.*
E-mail: {tian, shah}@cs.ucf.edu.

Manuscript received 22 Jan. 1996. Recommended for acceptance by L. Shapiro. For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 105221.

over M frames. The set of feature points associated with a single object lies in a four- (or less) dimensional subspace of the $2M$-d space of column vectors. Thus, the task of segmentation becomes identifying these subspaces and the column vectors that lie in them. This corresponds to finding reduced row echelon form of **W**, by applying Gauss-Jordan elimination to the rows of **W** with partial pivoting, or QR reduction with column pivoting [7]. In [6], a "shape interaction matrix," **Q**, is constructed as $\mathbf{V}\mathbf{V}^T$. The element of **Q**, $\mathbf{Q_{ij}}$, is zero only when feature trajectories i and j belong to different objects. Thus, segmentation is performed by putting **Q** in the block-diagonal form, where each block represents a moving object.

2.2 Perspective Projection

Torr and Murray [21] proposed a stochastic approach to segmentation using Fundamental Matrix (**F**) encapsulating the epipolar constraint. The method begins with generating hypothetical clusters by randomly sampling a subset of feature-correspondences and using them to calculate a solution, **F**. All feature-pairs consistent with the solution are included in the cluster by a *t-test*. Solutions are then pruned or merged. Finally, using an *integer programming* technique, the clusters are partitioned to form a segmentation. MacLean et al. [16] used mixture models to model multiple motion processes and applied EM algorithm to segmentation from linear constraints on 3D translation and bilinear constraints on 3D translation and rotation [13]. The results of clustering from the linear constraints are then used as an initial guess for parameter fitting using bilinear constraints. It is difficult to determine the number of clusters and the initial parameters for the EM algorithm. Thus, in our work, we determine the dominant object first, eliminate the object from subsequent analysis, and then repeat the same process on the remaining regions to find other objects. No prior knowledge of number of clusters and initial estimates is required.

Some researchers [3], [1], [5] suggested applying the Hough transform to motion estimation and segmentation. The advantages of the Hough transform are that it is relatively insensitive to noise and more robust, being a global approach, and the multiple local maxima in the parameter space naturally correspond to independently moving objects. In Ballard and Kimball's method [3], the 5D parameter space of translations and rotations is sampled, and each optical flow vector votes for all the consistent solution tuples. The tuples which receive maximal votes are taken as solutions. The method requires searching the candidate solutions over the entire five-dimensional parameter space, and also requires known depth. Using Hough transform on affine model, Adiv [1] identified regions in the image where optical flow is consistent with the movement of a planar surface. Then, he grouped these regions according to the consistency for various 3D motions. In general, a method not requiring planar surfaces is more useful. Bober and Kittler [5] exploited Hough transform and robust estimation to estimate optical flow and segment flow field based on an affine model. In our approach, we search only two-dimensional translation space, and extend adaptive Hough transform to compute 3D motion of multiple objects.

3 Motion Estimation Using Hough Transform

In this section, we describe a technique for computing 3D relative motion for each moving object given an optical flow field. Since we allow both camera and object motion, the effective motion for each object is the relative motion between the camera and the moving object.

Let $P = (X, Y, Z)$ denote a scene point in a camera-centered coordinate system, and let (x, y) denote the corresponding image coordinates. The image plane is located at $Z = f$ (the focal length). Under perspective projection point $P = (X, Y, Z)^t$ projects to $p = (x, y)^t$ in the image plane,

$$x = f\frac{X}{Z}, \quad y = f\frac{Y}{Z}. \tag{1}$$

The scene point P moves relative to camera with translation $\mathbf{T} = (T_x, T_y, T_z)^t$ and rotation $\mathbf{\Omega} = (\Omega_x, \Omega_y, \Omega_z)^t$. We assume that $\mathbf{T} \neq 0$, otherwise, depth Z cannot be determined. The relative motion of P can be expressed as

$$\frac{dP}{dt} = \mathbf{\Omega} \times + \mathrm{P}\mathbf{T}. \tag{2}$$

The optical flow $(u, v)^t$ of an image point (x, y) can be expressed as [15]:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \dfrac{fT_z - xT_z}{Z} - \dfrac{y}{f}\Omega_x + \left(f + \dfrac{x^2}{f}\right)\Omega_y - y\Omega_z \\ \dfrac{fT_y - yT_z}{Z} - \left(f + \dfrac{y^2}{f}\right)\Omega_x + \dfrac{xy}{f}\Omega_y + x\Omega_z \end{bmatrix}. \tag{3}$$

Since depth, Z, and translation, **T**, can be determined only up to a scale factor, we only solve for the translation direction and relative depth. Let the scale factor $s = \|\mathbf{T}\|$, speed. Consequently, we now let **T** denote a unit vector for translation direction and let Z_s denote the relative depth, Z/s. Unit vector **T** can be represented by spherical coordinates in terms of slant, θ, and tilt, ϕ: $(\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)^t$. Only half of the sphere must be considered, since solutions on the front and back halves are the same. Consequently, θ varies from 0^o to 90^o, and ϕ varies from 0^o to 360^o. There are five unknowns $(\theta, \phi, \Omega_x, \Omega_y, \Omega_z)$ associated with each moving object, and one unknown (Z_s) associated with each image point. We can eliminate depth Z from (3 and obtain:

$$\mathbf{c}(\mathbf{T})\mathbf{\Omega} = \mathrm{q}(\mathbf{T}), \tag{4}$$

where

$$\mathbf{c}(\mathbf{T}) = [fT_z x + T_y xy - T_x(f^2 + y^2),\ fT_z y + T_x xy - T_y(f^2 + x^2),\ fT_x x + fT_y y - (x^2 + y^2)T_z],$$

$$q(\mathbf{T}) = -fT_x v + fT_y u + T_z(xv - yu).$$

We collect N equations of (4) into the matrix form:

$$\mathbf{C}(\mathbf{T})\mathbf{\Omega} = \mathbf{q}(\mathbf{T}), \tag{4}$$

where

$$\mathbf{C}(\mathbf{T}) = \begin{bmatrix} \mathbf{c}_1(\mathbf{T}) \\ \mathbf{c}_2(\mathbf{T}) \\ \vdots \\ \mathbf{c}_N(\mathbf{T}) \end{bmatrix}, \quad \mathbf{q}(\mathbf{T}) = \begin{bmatrix} q_1(\mathbf{T}) \\ q_2(\mathbf{T}) \\ \vdots \\ q_N(\mathbf{T}) \end{bmatrix}.$$

At least three image points are needed to solve for the rotation parameters in $\mathbf{\Omega}$. We compute a least-squares estimate of rotation for a fixed choice of **T**:

$$\mathbf{\Omega} = (\mathbf{C}(\mathbf{T})^t\mathbf{C}(\mathbf{T}))^{-1}\mathbf{C}(\mathbf{T})^t\mathbf{q}. \tag{6}$$

In order to deal with multiple moving objects, we partition the entire image into patches. Within each patch, we compute a least-squares estimate of the rotation (6) for a given sample of **T**, and count the corresponding vote for a sample in the five-dimensional parameter space. Within the framework of the standard Hough transform, instead of evaluating the entire five-dimensional parameter space, we only examine the two-dimensional translational parameter space from which we compute the corresponding optimal solution for three rotation parameters.

In a scene containing multiple moving objects, some image patches contain multiple motions. When estimating 3D motion pa-

rameters, the larger the field of view, the more accurate the estimates. The larger patches are, therefore, less sensitive to noise, but more likely to contain multiple motions (in our experiments, we used 35×35 patches). The least-squares estimation is computationally efficient, but not robust, particularly for multiple motions.

4 Robust Motion Estimation of Multiple Objects

In this section, we present a robust method for multiple motion estimation. Multiple motions within a patch can be treated as outliers with respect to a dominant motion. M-estimators are able to handle outliers and Gaussian noise in optical flow measurements simultaneously, so, we use a redescending M-estimator in our scheme. For a fixed $\mathbf{T}$, the rotation estimate of a dominant motion within a patch is estimated using an M-estimator; the other minor motions are rejected as outliers.

The M-estimators minimize the sum of a symmetric, positive-definite function $\rho(r_i)$ of the residuals r_i, with a unique minimum at $r_i = 0$. ρ functions have been designed to reduce the influence of the large residual values of the estimated fit. The influence function, ψ, is defined as the derivative of ρ, $\psi(r_i) = \frac{d\rho(r_i)}{dr_i}$. When the residuals are large, the ψ increases with deviation, then starts decreasing to zero, so that very deviant points—the true outliers—are not used in the estimation of the parameters. There are several possible choices for ρ function listed in [8]. In problems of nonlinear model fitting (e.g., [4]), where initial estimates are not available, M-estimators (e.g., Lorentzian estimator) can be desirable, whose influence functions are continuous and "redescend" to nearly zero outside the central region, while all data points affect the estimation. But M-estimators (e.g., Beaton and Tukey's estimator), whose influence functions beyond a certain threshold reduce to zero, are inappropriate for the problems requiring initial estimates. In our problem, however, since we fit a linear model (4), which does not require initial estimates, both of the above M-estimators methods are applicable. In our implementation, Beaton and Tukey's biweight function is used:

$$\rho(r) = \begin{cases} \left(C_B^2/2\right)\left[1 - \left[1 - \left(r/C_B\right)^2\right]^3\right] & \text{if } |r| \le C_B \\ C_B^2/2 & \text{otherwise,} \end{cases}$$

where r is residual, and C_B is a turning constant. Holland and Welsch [10] recommended $C_B = 4.685$ to achieve superior performance for Gaussian noise. Since we are dealing with the patch which may contain multiple motions, a smaller turning constant should be used.

M-estimation problems are usually solved using an iterative weighted least-squares method [10], in which a weight is computed for each data point based on the residual error of the previous estimate. Initially, the weights are all one. After the vector $\mathbf{\Omega}$ (denoted by $\hat{\mathbf{\Omega}}_0$) is computed with the contribution of all data points in the patch, the weights are updated according to the following:

$$\mathrm{w}(r) = \frac{1}{r}\frac{d\rho(r)}{dr} = \begin{cases} \left[1 - \left(r/C_B\right)^2\right]^2 & \text{if } |r| \le C_B \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

The vector $\mathbf{\Omega}$ is refined through iterations:

$$\hat{\mathbf{\Omega}}_1 = \hat{\mathbf{\Omega}}_0 + \left(\mathbf{C}^{\mathrm{t}} < \mathrm{w}\left(\frac{\mathbf{q} - \mathbf{C}\hat{\mathbf{\Omega}}_0}{\sigma}\right) > \mathbf{C}\right)^{-1} \mathbf{C}^{\mathrm{t}} < \mathrm{w}\left(\frac{\mathbf{q} - \mathbf{C}\hat{\mathbf{\Omega}}_0}{\sigma}\right) > \left(\mathbf{q} - \mathbf{C}\hat{\mathbf{\Omega}}_0\right),$$

where $< >$ denotes an $N \times N$ diagonal matrix, and σ is a scale parameter which can be estimated by

$$\sigma = 1.4826\, med\ |\, r_i - med\ r_i\ |, \quad (8)$$

where i denotes feature i and med denotes the median taken over the entire patch, $\sigma \neq 0$.

However, M-estimators can only allow at most $1/(m + 1)$ of contamination for the data, where m is the number of parameters in the least-squares estimation. To reduce this effect, we use the scheme of dividing image into overlapping patches. We use the following measure to determine the convergence of the algorithm:

$$E^{(l)} = \sqrt{\frac{\sum_{i=1}^{n} w_i r_i^2}{\sum_{i=1}^{n} w_i}},$$

where l denotes the iteration number, and n denotes the number of nonzero weights corresponding to the number of inliers, which contribute to the robust estimate.

5 Adaptive Hough Transform

The Hough transform involves representing the continuous parameter space by an appropriately quantized discrete space. The fineness of quantization is crucial to the accuracy in the determination of parameters. It also requires the identification of significant local maxima in the number of votes within the accumulator array. Using a large accumulator array is not practical in many respects. In our problem, if translation parameter ranges are quantized to 0.5° per interval and each rotation parameter range of $(-6^{\circ}, 6^{\circ})$ is quantized to 0.1° per interval, $O(10^{11})$ ($720 \times 180 \times 120 \times 120 \times 120$) elements are required for storage. A large accumulator array also requires large computations, since it requires that many parameter cells have to be tested, and this huge array has to be searched to locate local maxima. Even searching for the 2D translation parameters would require testing $O(10^5)$ cells. In this section, we describe the technique to reduce these computational and storage burdens by using Adaptive Hough transform (AHT). We extended the original AHT proposed by Illingworth and Kittler [11] to deal with this particular five-dimensional parameter space.

The AHT uses a small accumulator array and iterative "coarse-to-fine" accumulation and search strategy to identify significant peaks in Hough parameter space. The technique begins with the coarsest quantization of the original parameter range, accumulates the HT in a small size accumulator array, and uses this information to refine the parameter range so that interesting areas can be investigated in greater detail through the finer quantization. The process continues until parameters are determined to a prespecified accuracy. The located parameters are used to identify the object moving with these motion parameters. Then, a search for another object is initialized at the coarsest resolution in the remaining image regions. Images containing multiple objects can, therefore, be processed until the parameter space contains no significant structure.

In this work, we chose a $9 \times 9 \times 9 \times 9 \times 9$ accumulator array to provide enough samples of parameter space and also to keep the accumulator array as small as possible. Adaptively searching for the 2D translation parameters requires testing $O(10^2)$ cells. The crucial part is to reset the parameter range in the vicinity of significant peaks for the next iteration. The detection of significant maxima can be achieved using a scheme which binarizes the accumulator array and labels connected components in this binary array. Since it is difficult to sequentially compute connected components in 5D space, we chose to compute only the connected components in the corresponding 2D translational accumulator array. To redefine the translation range, we first combined the votes from rotation space corresponding to each translation sample, and

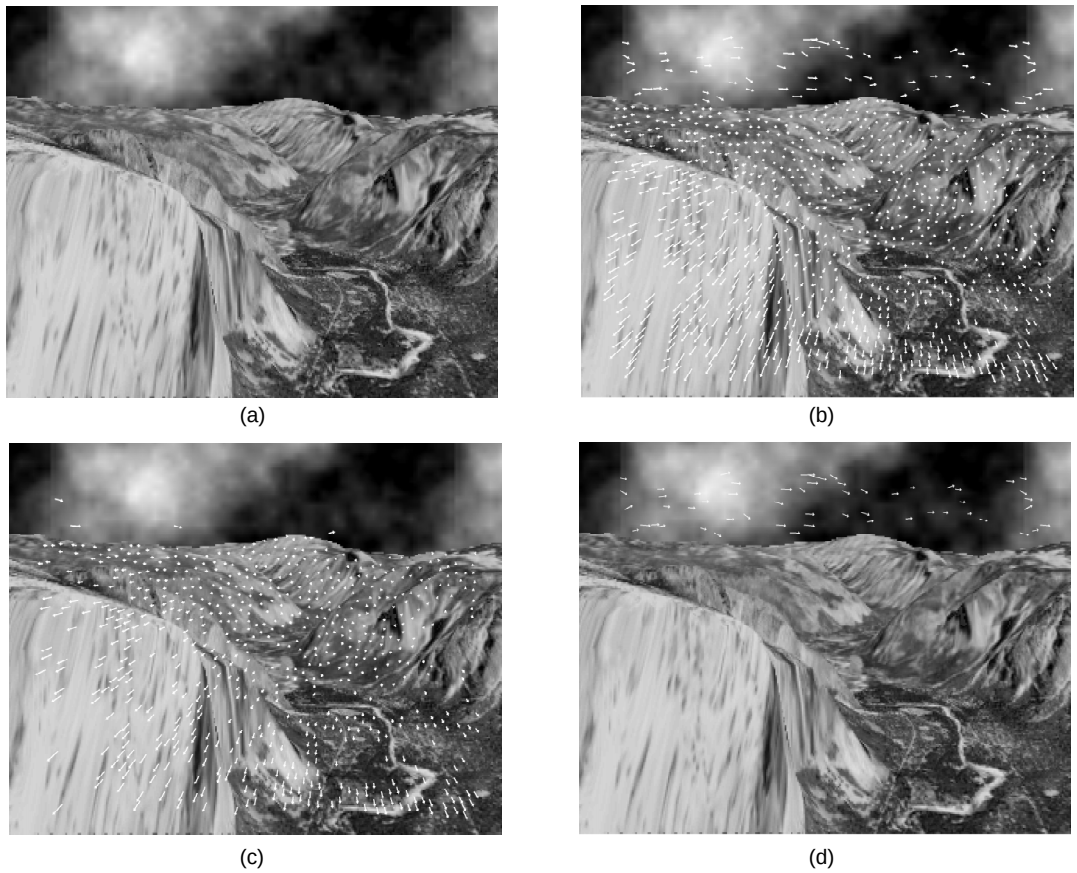

Fig. 1. Results for Yosemite sequence with camera and clouds motion. (a) Frame 1. (b) Measured optical flow field overlay with Frame 1. The vector lengths are scaled by 2.0. (c) Segmented Yosemite Valley. Four features of clouds were incorrectly segmented into Yosemite Valley, due to the ambiguity between depth and 3D translation in a monocular sequence. They can be eliminated from the segment of Yosemite Valley due to their negative depth estimates. (d) Segmented clouds.

then used a connected component algorithm [14] to determine the parameter range which produces maximal votes. The actual translation parameters must exist within this range. The criteria used to identify the best peak from the set of connected components is the density of the connected component, i.e., number of votes / number of bins in the 5D accumulator array. The parameter range is adjusted according to the location and extent of the best peak. Once the translation parameter range is redefined, the rotation parameter range is also adjusted to a new extent at a finer resolution, corresponding to the new translation range. This process is repeated until the adjusted translation parameter ranges are very small. Then, the sample point in the current 5D parameter space having the maximal vote determines the motion parameters. The 3D rigidity constraint contained in (4) is used to identify the image points which belong to the corresponding moving object.

6 Experimental Results

This section describes experiments using synthetic images and real images. In practice, optical flow cannot be reliably computed for each pixel of the entire image due to the noise in the image and the "aperture problem." Thus, we only rely on the optical flow in the textured regions. We measured the flow field using Tomasi and Kanade's algorithm [19]. In our experiments, entire images were divided into overlapped patches of 35×35 to provide a relatively large field of view for motion estimation within each patch.

6.1 Synthetic Images

Fig. 1a shows a frame of a computer graphics generated sequence of a flight through Yosemite Valley. In this sequence, both the camera and clouds are moving. Fig. 1b shows the flow field computed from two consecutive frames. This optical flow field was used as input to recover 3D motion parameters, and to segment the scene. The actual translation direction was $\mathbf{T} = (0.0077, -0.1828, 0.9831)$, and the actual speed of motion was 34.7 (in unit of pixels). The actual rotation axis was (0.0250, 0.9798, 0.1985) and rotation rate was 0.0961^{o}/frame. Our method recovered the motion parameters with 0.46^{o} of the translation error, 4.24^{o} of the rotation axis error, and 0.003^{o} of the rotation rate error. The error in motion estimates is due to the error in optical flow estimates and quantization of the motion parameter space. Fig. 1c shows the segment of Yosemite Valley using the recovered 3D motion parameters, where four features of clouds were incorrectly segmented into Yosemite Valley, due to the ambiguity between depth and 3D translation in a monocular sequence. This ambiguity can be resolved by imposing the *positive depth constraint*, which implies that each visible point lies in front of the camera. Depth values for Yosemite Valley were computed using recovered motion parameters. The recovered depth values were close to true depth values. Fig. 1d shows the remaining image region, clouds segment. In this sequence, the clouds in the background change their shape over time; the clouds undergo a nonrigid motion. Thus, we make no attempt to solve for 3D motion of the clouds.

We compared the performance of our algorithm with other egomotion methods. Table 1 summarizes the errors in the motion parameters estimated by E-matrix method [23], subspace method [9], and the proposed adaptive Hough transform (AHT) method for Yosemite sequence. Our algorithm performs better because we explicitly take the outliers of the flow estimates into account, so that only the reliable flow estimates contribute to 3D motion estimation.

(a) (b) (c)

Fig. 2. An egomotion sequence in which the camera is translating and rotating in front of a static planar surface. (a) Frame 1. (b) Frame 2. (c) Measured optical flow field overlay with Frame 1. The vector lengths are scaled by 2.0.

(a) (b)

(c) (d)

Fig. 3. Results for a sequence with camera and right toy house moving. (a) Frame 1. (b) Measured optical flow field overlay with Frame 1. The vector lengths are scaled by 2.0. (c) Segmentation result for Object 1: the background. (d) Segmentation result for Object 2: the right toy house.

TABLE 1
ESTIMATION ERRORS WITH E-MATRIX, SUBSPACE, AND AHT METHOD FOR YOSEMITE SEQUENCE WITH UNIFORM BACKGROUND

Method	Translation error	Rotation axis error	Rotation rate error
E-matrix	4.8^o	25^o	0.016^o
Subspace	3.5^o	44^o	0.15^o
AHT	0.46^o	4^o	0.003^o

The translation error was measured by the angle between the actual and recovered translation direction. The rotation error was measured by the rotation axis error (i.e., the angle between the actual and recovered rotation axis) and the rotation rate error.

6.2 Real Images

The real images in this section were taken by a SONY 25-mm CCD camera (field of view is about 30^o). The focal length and aspect ratio of the camera are calibrated, but no effort has been made to correct the geometric distortion.

6.2.1 Egomotion

Figs. 2a and 2b show two frames of a sequence in which the camera is moving in front of a static planar surface. Fig. 2c shows the flow field computed from two consecutive frames. The actual translation direction was $\mathbf{T} = (0, 0, 1)$, and the actual rotation was $(0^o, -0.2^o, 0^o)$. The recovered translation direction was (0.02, 0, 0.9998), which has error of 1.23^o. The estimated rotation was $(0.02^o, -0.22^o, 0.01^o)$. The angle between actual and recovered rotation axis was 5.8^o, and rotation rate error was 0.02^o. It can be observed that the measured flow field for this sequence contains a number of outliers due to noise in the input images. However, our method can still robustly recover the motion parameters, rejecting these outliers.

6.2.2 Two Moving Objects

Fig. 3a shows a frame of a sequence in which both the camera and the right toy-house are moving. The camera is translating towards the scene along z-axis, and rotating around y-axis by 0.1^o/frame. The toy-house is translating to the left and rotating around y-axis in the opposite direction to the camera rotation. Fig. 3b shows the measured flow field. The actual translation direction of the camera was $\mathbf{T} = (0, 0, 1)$, and the recovered translation direction was (0, 0, 0.999997), with error of 0.13^o. The estimated rotation of camera was $(0^o, -0.1^o, 0.009^o)$. The recovered effective translation direction and rotation of the right toy-house was (0.17, 0, 0.985) and $(0^o, 0.3^o, -0.03^o)$, respectively. Here, we recovered the relative motion between the camera and the moving object. Figs. 3c and 3d show the segmentation results using the recovered 3D motion parameters.

7 DISCUSSION

In this paper, we have proposed a method to determine 3D motion for multiple objects from two perspective views, using adaptive

Hough transform and robust estimation techniques. Segmentation is determined based on a 3D rigidity constraint. We explicitly take the outliers into account, so that only the reliable flow estimates contribute to 3D motion estimation, thus, our method can robustly recover 3D motion parameters, rejecting the outliers in optical flow estimates. Applications of the proposed method to both synthetic and real image sequences have been demonstrated with promising results. Our implementation of the algorithm in C can be accessed at ftp://eustis.cs.ucf.edu/.

There are some limitations of this algorithm. First, in AHT, to detect significant maxima of votes in accumulator array, we compute only the connected components in the corresponding 2D translational accumulator array, by first combining the votes from rotation space corresponding to each translation sample (since it is difficult to sequentially compute connected components in 5D space). Since segmentation is performed solely based on translation direction, the algorithm fails when objects move in the same translation direction. Second, some heuristics are introduced to identify the best peak from the set of connected components. This may involve parameter tuning, otherwise, algorithm may get into local minimum. Third, (2) is an instantaneous approximation for small rotations. It is shown in [9] that the errors introduced by the instantaneous approximation are quite small when the rotation rate is less than three degrees. When rotation rate is large, discrete-time motion model is more accurate.

ACKNOWLEDGMENTS

We would like to thank Prof. David Heeger for helpful discussions. The research reported here was supported by U.S. National Science Foundation grants CDA-9122006 and IRI-9220768.

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, pp. 384-401, 1985.

[2] S. Ayer, P. Schroeter, and J. Bigun, "Segmentation of Moving Objects by Robust Motion Parameter Estimation over Multiple Frames," *ECCV*, pp. 316-327, 1994.

[3] D. Ballard and O. Kimball, "Rigid Body Motion from Depth and Optical Flow," *Computer Visualization, Graphics, and Image Processing*, vol. 22, pp. 95-115, 1983.

[4] M.J. Black and Y. Yacoob, "Tracking and Recognizing Rigid and Non-Rigid Facial Motions Using Local Parametric Models of Image Motion," *ICCV*, pp. 374-381, 1995.

[5] M. Bober and J. Kittler, "Estimation of Complex Multimodal Motion: An Approach Based on Robust Statistics and Hough transform," *Image and Vision Computing*, vol. 12, pp. 661-668, 1994.

[6] J. Costeira and T. Kanade, "A Multi-Body Factorization Method for Motion Analysis," *ICCV*, pp. 1,071-1,076, 1995.

[7] C.W. Gear, "Feature Grouping in Moving Objects," *Proc. IEEE Workshop Motion of Non-Rigid and Articulated Objects*, pp. 214-219, 1994.

[8] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel, *Robust Statistics: An Approach Based on Influence Function*. New York: Wiley, 1986.

[9] D. Heeger and A. Jepson, "Subspace Methods for Recovering Rigid Motion I: Algorithm and Implementation," *Int'l J. Computer Visualization*, vol. 7, pp. 95-117, 1992.

[10] P. Holland and R. Welsch," Robust Regression Using Iteratively Reweighted Least Squares," *Comm. Statistics—Theoretical Method*, pp. 813-827, 1977.

[11] J. Illingworth and J. Kittler, "Adaptive Hough transform," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, pp. 690-698, 1987.

[12] M. Irani, B. Rousso, and S. Peleg, "Detecting and Tracking Multiple Moving Objects Using Temporal Integration," *ECCV*, pp. 282-287, 1992.

[13] A.D. Jepson and D.J. Heeger, "Linear Subspace Methods for Recovering Translation Direction," *Spatial Vision in Humans and Robots*, L. Harris and M. Jenkin, eds., pp. 39-62. New York: Cambridge Univ. Press, 1993.

[14] R. Lumia, L. Shapiro, and O. Zuniga, "A New Connected Component Algorithm for Virtual Memory Computers," *Computer Visualization, Graphics, and Image Processing*, vol. 22, pp. 287-300, 1983.

[15] H.C. Longuet-Higgens and K. Prazdny, "The Interpretation of a Moving Retinal Image," *Proc. Royal Soc. of London*, B 208, 1980.

[16] W.J. MacLean, A.D. Jepson, and R.C. Frecker, "Recovery of Egomotion and Segmentation of Independent Object Motion USING the EM Algorithm," *Proc. British Machine Vision Conf.*, pp. 13-16, York, U.K., 1994.

[17] W. Thompson, P. Lechleider, and E. Stuck, "Detecting Moving Objects Using the Rigidity Constraint," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 162-166, 1993.

[18] W. Thompson, K. Mutch, and V. Berzins," Dynamic Occlusion Analysis in Optical Flow Fields," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 7, pp. 374-383, 1985.

[19] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features, Shape and Motion from Image Streams: A Factorization Method—Part 3," Technical Report CMU-CS-91-132, Carnegie Mellon Univ., 1991.

[20] C. Tomasi and T. Kanade, "Shape from Motion from Image Streams under Orthography: A Factorization Method," *Int'l J. Computer Visualization*, vol. 9, pp. 137-154, 1992.

[21] P. Torr and D. Murray, "Stochastic Motion Clustering," *ECCV*, pp. 328-337, 1994.

[22] J. Wang and E. Adelson, "Layer Representation for Motion Analysis," *Computer Visualization and Pattern Recognition*, pp. 361-366, 1993.

[23] J. Weng, T. Huang, and N. Ahuja, "Motion and Structure from Two Perspective Views: Algorithms, Error Analysis, and Error Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 451-476, 1989.