Proc. IEEE ICIP (International Conference on Image Processing), Rochester NY, September 2002, pp. III-513 -- III-516

VANISHING POINTS AND 3D LINES FROM OMNIDIRECTIONAL VIDEO

Michael Bosse, Richard Rikoski, John Leonard, Seth Teller

Massachusetts Institute of Technology,
Cambridge MA 02139, USA
{ifni,rikoski,jleonard,seth}@mit.edu

Abstract

This paper describes a system for structure-from-motion using vanishing points and three-dimensional lines extracted from omni-directional video sequences. Two novel aspects of this work are its deferred initialization of features using noisy observations from multiple, uncertain vantage points, and its use of dynamic programming for efficient 3D line tracking. We show preliminary results from the system for both indoor and outdoor sequences.

1. INTRODUCTION

There is increasing interest in the development of structure from motion (SFM) algorithms capable of running in real-time [5, 6]. Real-time SFM will enable applications such as (1) real-time navigation of mobile robots in unknown environments, (2) real-time capture of 3-D computer models using hand-held cameras, and (3) real-time head tracking in extended environments.

This paper presents a system that uses vanishing points (VPs) and 3-D line segments as features in a stochastic framework for recursive SFM. The approach assumes that the scene contains sets of stationary parallel lines, a valid assumption in many human-made environments.

Concurrent recovery of scene structure and camera trajectory is a high-dimensional, coupled state estimation problem. The key challenges here include coping with uncertainty and scale, and the coupling (non-independence) of errors in feature and camera pose estimates. This paper uses the extended Kalman filter (EKF) for recursive state estimation. Our use of VPs for accurate rotation estimation effectively sidesteps two limitations of the EKF: its potential for divergence when angular error is large, and its inability to handle multi-modal distributions. Our choice of features, and conservative approach to feature initialization and matching, greatly eases the data association problem. This paper does not address the problems of large loop-closing and global relocalization [7, 8].

An important objective of our work is to tie estimated scene structure to a common reference frame defined by the initial camera pose, as in the work in robotics known as simultaneous localization and mapping (SLAM) [9, 10, 11, 8], using laser range scanners [10, 12, 11, 13]. Some vision researchers have pursued similar approaches for limited scenes [5, 14].

This research has been funded in part by NSF Career Award BES-9733040, the MIT Sea Grant College Program under grant NA86RG0074 (project RCM-3), the Office of Naval Research under grant N00014-97-0202, and by Draper Laboratories under contracts DL-H-516617, DL-H-526716, and DL-H-539054.

2. THE ALGORITHM

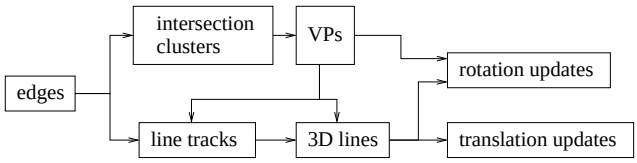

Fig. 1. Data Flow Graph.

Figure 1 summarizes the data flow in our system. Given a sequence of omni-directional images and detected linear features, our task is to estimate the 3D position and orientation of scene landmarks (VPs and 3D lines), and the pose of the camera as each image was acquired. We join uncertain landmark states and robot poses in a common state vector, with a corresponding error covariance matrix:

$$\vec{x}_{\text{world}} := \begin{bmatrix} \vec{x}_{\text{robot}} \\ \vec{x}_{\text{map}} \end{bmatrix} \quad (1)$$

$$\vec{P}_{\text{world}} := \begin{bmatrix} \vec{P}_{\text{rr}} & \vec{P}_{\text{rm}} \\ \vec{P}_{\text{mr}} & \vec{P}_{\text{mm}} \end{bmatrix} \quad (2)$$

The state projection function models the motion of the camera. The measurement prediction function models landmark observations. Newly discovered landmarks ($\vec{y}$) are incorporated into the map from the measurements ($\vec{z}$), using an initialization function $g(\cdot)$. The initialization function augments both the state vector and covariance matrix, ex-

plicitly correlating the new landmarks with all other landmarks:

$$\vec{y} = \vec{g}(\vec{x}, \vec{z}) \quad (3)$$

$$\vec{x} \leftarrow \begin{bmatrix} \vec{x} \\ \vec{y} \end{bmatrix} \quad (4)$$

$$\vec{P}_{xx} \leftarrow \begin{bmatrix} \vec{P}_{xx} & \vec{G}_x \vec{P}_{xx} \\ \vec{P}_{xx}\vec{G}_x^T & (\vec{G}_x \vec{P}_{xx} \vec{G}_x^T + \vec{G}_z \vec{P}_{zz} \vec{G}_z^T) \end{bmatrix} \quad (5)$$

where $g(\cdot)$, $\vec{G}_x$, and $\vec{G}_z$, are the mapping function and its Jacobians, and $\vec{P}_{zz}$ is the measurement covariance.

Reobservation of a landmark improves the estimate of the camera pose, the observed landmarks, and any correlated landmarks. Cross-covariances determine the degree to which correlated states change in the presence of new information, and propagate information to features that are not observed in the current frame.

Three-dimensional features may not be fully observable from a single vantage point. Thus, our method combines observations from multiple vantage points [15], by retaining several recent, correlated estimated camera positions. This technique also makes the filter more robust, providing a "probationary period" before measurements are integrated into the EKF and hence affect the SFM computation. This approach loses no information, since data from the probationary period is eventually fully incorporated into the solution.

2.1. Estimating Vanishing Points

Vanishing points (VPs) are the common directions of parallel 3-D lines [16]. In a perspective view, the VP is at the intersection of the images of the lines. We detect potential VPs in each view using RANSAC, and refine the estimated VP directions using EM (expectation-maximization) [17]. EM generates a classification of observed lines to modeled VPs, and a direction estimate for each VP. Since true VPs are at infinity, they are invariant to translation, whereas local features are not. Therefore, we delay the initialization of VPs until the camera moves. Once a VP is added to the state vector, the EKF is updated with all past views of the VP using the retained brief history of saved camera pose states.

VPs are represented as 3D unit vectors in world coordinates, with two effective degrees of freedom (DOFs). Thus the uncertainty in VP parameters lies on the unit sphere. We linearize the constraint at the VP to be the tangent plane of the unit sphere with its normal equal to the VP.

2.2. Tracking Image Lines

Lines are tracked across consecutive image frames using stochastic nearest-neighbor gating [18] augmented by a novel ordering constraint. The relative order of parallel lines on a single surface persists as the camera moves (some lines may disappear or reappear due to occlusion). We exploit this fact to develop an efficient 3D line tracker using dynamic programming. We project newly observed lines into the current view, then sort them around each corresponding VP. We next apply a modified version of the longest common substring algorithm [19] to find the best match to previously observed lines, while maintaining the ordering constraint. The mean colors to the left and right of the line are used to further reduce the chance of false matches.

2.3. Mapping 3-D Lines

In addition to VPs, the system tracks local, parallel 3D line segments that share a common VP. These segments have six DOFs, which we partition into three groups: two for the direction of the line, two for the perpendicular offset of the line from the origin, and two for the distances of the endpoints along the line (these are not included in the EKF state). The system updates 3D lines by projecting them into the current view, and comparing them with the corresponding line extracted from the current view.

3. EXPERIMENTAL RESULTS

We present experimental results using two omni-directional video sequences. The experimental system consists of a digital video camcorder attached to a parabolic mirror which was oriented vertically for the indoor sequence and horizontally for the outdoor sequence. For the indoor sequence, the camera was mounted on a small mobile robot as shown in Figure 2(a). To maintain the calibration of the omni-camera despite vibration during data acquisition, fiducial marks adjacent to the mirror (visible in Figure 2) were tracked. For synchronization, we used a software modem to encode timestamps from the robot in the camcorder's audio channel.

The indoor sequence consisted of 2,400 frames with a trajectory length of 106 meters. Odometry data from the robot was used for EKF state projection. The error drift rate is approximately 15 degrees per minute. The robot moved at a speed of approximately 25 centimeters per second. Sample images and features are shown in Figure 2. Laser scanner data and commercial robot navigation software provided ground-truth camera pose with an accuracy of approximately 5 centimeters and 0.5 degrees.

Figure 3 shows a comparison of the camera trajectory as estimated by the SFM algorithm with odometry and with the output from a two-dimensional (three DOF) laser-gyro navigation system. Comparisons with ground-truth are only possible about one rotation axis (yaw), however our algorithm computes a full six DOF solution for the camera trajectory. Figure 4 shows the distribution of rotational and

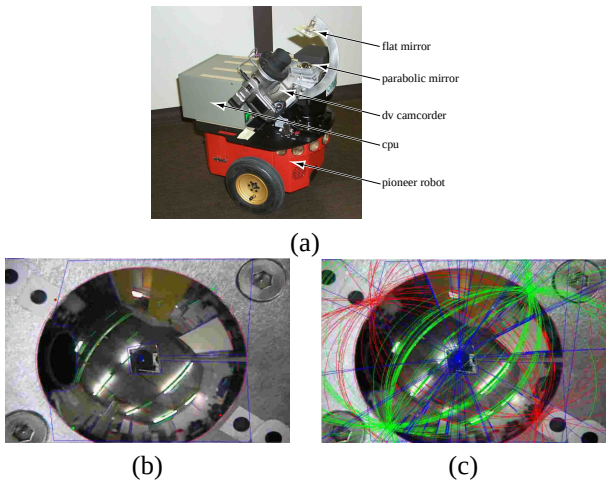

Fig. 2. (a) The omni-directional video camera mounted on a Pioneer robot. (b) Edges and VPs extracted. (c) Edges extended to infinite lines.

translation errors of the visual SFM output as compared to the laser/gyro estimates. The standard deviation of heading errors was one degree, and the standard deviations for translation errors were 30 centimeters in x and y and 6 centimeters in z. Figure 5 shows the distribution of vanishing points referenced to odometry and to the visual SFM output. This demonstrates the method's ability to decouple estimation of rotational and translational errors. Finally, Figure 6 shows two views of the map estimated by the algorithm, compared with a 2-D reference laser map.

The outdoor sequence consisted of 17,000 frames with a path length of 946 meters. Figure 7 shows the estimated map and camera trajectory, compared to odometry. (No ground-truth is available for this sequence.)

The current implementation of the algorithm uses MATLAB and has not been optimized for speed. Processing times are roughly between 1 and 3 frames per second. We are developing an optimized C++ implementation to enable real-time operation.

4. CONCLUSION AND FUTURE WORK

This paper described a new method for recursive SFM from omni-directional video sequences. Vanishing points and 3-D lines are used as features in a recursive state estimation framework [9]. The approach has been demonstrated with off-line processing of both indoor and outdoor image sequences. In the indoor experiment, the resulting trajectory estimate and the map estimated from the omni-directional video data compare favorably with a trajectory estimate and map generated from laser scanner data.

Several key issues must be addressed in the future to ex-

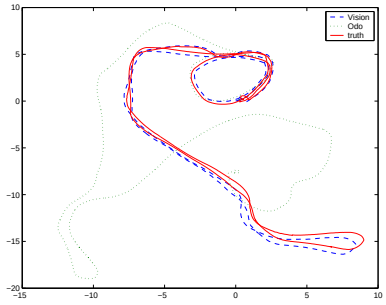

Fig. 3. Comparison of trajectories estimated from the SFM algorithm, a commercial laser/gyro navigation system, and odometry for indoor experiment. The total path length was 106 meters and the robot returned to within approximately 30 centimeters of its starting point.

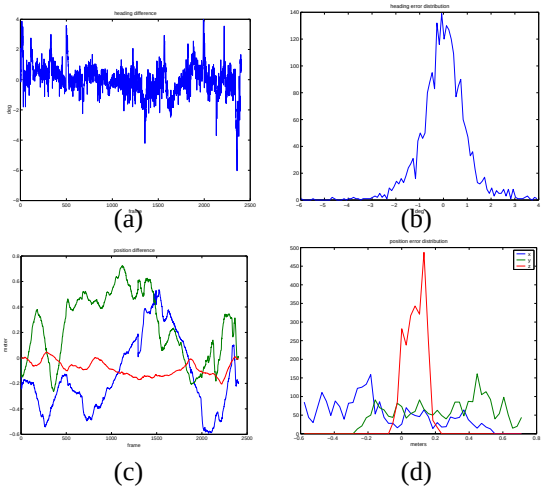

Fig. 4. State estimation errors. (a) Difference between heading estimated by the laser/gyro system and the SFM algorithm. (b) Histogram of heading errors (note that there is only a 1 degree standard deviation). (c) Difference between translation (x, y, z) estimated by the laser/gyro system and the SFM algorithm. (d) Histogram of translation errors.

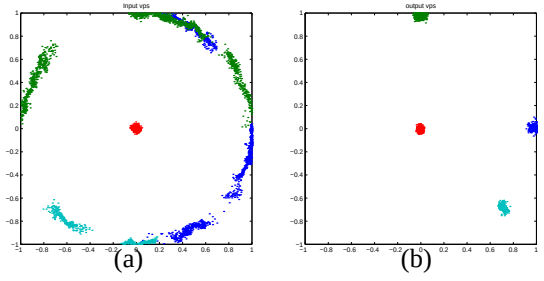

Fig. 5. Distribution of vanishing points: In (a) we see the vanishing points projected into the omni-directional image using the heading estimate from the odometry only. In (b) we see the VPs projected using the rotation estimated by our algorithm.

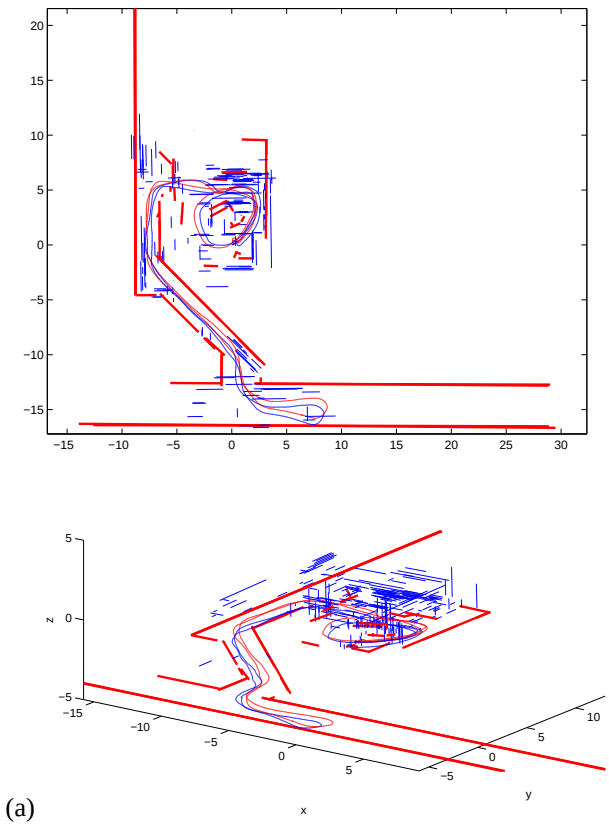

Fig. 6. 3-D wireframe model of map for indoor experiment. For comparison, walls estimated from laser data are shown in bold. (a) Bird's eye view. (b) Oblique view. Note: Line segments from the laser mapping system (shown as bold) represent walls that are projected onto the ground plane.

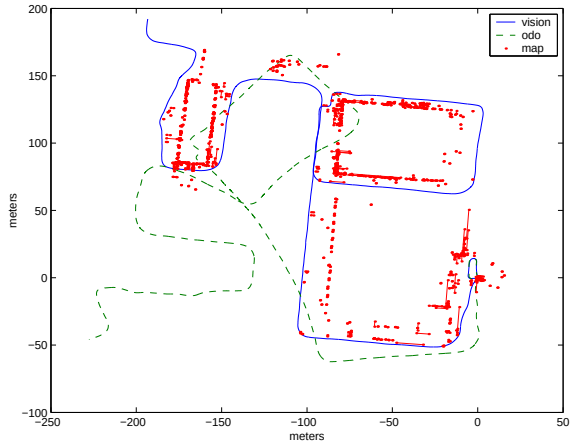

Fig. 7. Comparison of trajectories estimated from the SFM algorithm and odometry for outdoor experiment. The image sequence is 17,000 frames acquired over a 946-meter path.

tend this work to create a complete, real-time SFM system for human-made environments, including: (1) computationally efficient large-scale mapping; (2) loop-closing (i.e., submap matching); and (3) mapping of aggregate features.

5. REFERENCES

[1] Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521623049 (2001)

[2] Faugeras, O., Luong, Q.T., Papadopoulo, T.: The Geometry of Multiple Images. MIT Press (2001)

[3] Triggs, B., Zisserman, A., Szeliski, R., eds.: Vision algorithms, theory and practice: International Workshop on Vision. Springer-Verlag (1999)

[4] Taylor, C.J., Kriegman, D.J.: Structure and motion from line segments in multiple images. IEEE Trans. Pattern Analysis and Machine Intelligence **17** (1995) 1021–1032

[5] Davison, A.J.: Mobile Robot Navigation Using Active Vision. PhD thesis, University of Oxford (1998)

[6] Chiuso, A., Favaro, P., Jin, H., Soatto, S.: 3-d motion and structure from 2-d motion causually integrated over time: Implementation. In: Sixth European Conference on Computer Vision. (2000)

[7] Gutmann, J.S., Konolige, K.: Incremental mapping of large cyclic environments. In: International Symposium on Computational Intelligence in Robotics and Automation. (1999)

[8] Thrun, S.: An online mapping algorithm for teams of mobile robots. Int. J. Robotics Research **20** (2001) 335–363

[9] Smith, R., Self, M., Cheeseman, P.: A stochastic map for uncertain spatial relationships. In: 4th International Symposium on Robotics Research. MIT Press (1987)

[10] Moutarlier, P., Chatila, R.: An experimental system for incremental environment modeling by an autonomous mobile robot. In: 1st International Symposium on Experimental Robotics, Montreal (1989)

[11] Castellanos, J.A., Montiel, J.M.M., Neira, J., Tardos, J.D.: The SPmap: A probabilistic framework for simultaneous localization and map building. IEEE Trans. Robotics and Automation **15** (1999) 948–952

[12] Dissanayake, M.W.M.G., Newman, P., Durrant-Whyte, H.F., Clark, S., Csorba, M.: An experimental and theoretical investigation into simultaneous localization and map building. In: Sixth International Symposium on Experimental Robotics. (1999) 265–274

[13] Guivant, J., Nebot, E.: Optimization of the simultaneous localization and map building algorithm for real time implementation. IEEE Transactions on Robotic and Automation **17** (2001) 242–257

[14] McLauchlan, P.F.: A batch/recursive algorithm for 3d scene reconstruction. In: Int. Conf. Computer Vision and Pattern Recognition. Volume 2., Hilton Head, SC, USA (2000) 738–743

[15] Leonard, J.J., Rikoski, R.: Incorporation of delayed decision making into stochastic mapping. In Rus, D., Singh, S., eds.: Experimental Robotics VII. Lecture Notes in Control and Information Sciences. Springer-Verlag (2001)

[16] Antone, M., Teller, S.: Automatic recovery of relative camera rotations for urban scenes. In: Proc. CVPR. (2000) II–282–289

[17] Antone, M.E.: Robust Camera Pose Recovery Using Stochastic Geometry. PhD thesis, MIT (2001)

[18] Bar-Shalom, Y., Fortmann, T.E.: Tracking and Data Association. Academic Press (1988)

[19] Cormen, T., Leiserson, C., Rivest, R.: Introduction to algorithms. The MIT Press (1991)