The Visual Computer

Vanishing points and three-dimensional lines from omni-directional video

Michael Bosse,
Richard Rikoski,
John Leonard,
Seth Teller

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

Published online: ?? ?? 2003

This paper describes a system for structure from motion using vanishing points and three-dimensional lines extracted from omni-directional video sequences. To track lines, we use a novel dynamic programming approach to improve ambiguity resolution, and we use delayed states to aid in the initialization of landmarks. By reobserving vanishing points we get direct measurements of the robot's three-dimensional attitude that are independent of its position. Using vanishing points simplifies the representation since parallel lines share the same direction states. We show the performance of the system in various indoor and outdoor environments and include comparisons with independent two-dimensional reference maps for each experiment.

Key words: Omni-directional video – Vanishing points – Structure from motion – Visual navigation – Image line tracking

1 Introduction

There is increasing interest in the development of structure from motion (SFM) algorithms capable of running in real-time [6, 8]. Real-time SFM will enable applications such as (1) real-time navigation of mobile robots in unknown environments, (2) real-time capture of three-dimensional (3-D) computer models using hand-held cameras, and (3) real-time head tracking in extended environments.

The set of choices in the development of structure from motion algorithms include:

- state estimation (batch vs. recursive);
- choice of representation (geometric vs. appearance-based);
- choice of features (points, edges, lines, curves, etc.);
- camera geometry (monocular, binocular, trinocular, omni-directional; affine vs. projective, calibrated vs. uncalibrated cameras);
- representation and manipulation of uncertainty (Kalman filtering, sum of Gaussians, sequential Monte Carlo methods, etc.).

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. The approach is demonstrated experimentally with results from several long (1000+ frames) omni-directional video sequences.

A critical decision in algorithm development is the choice between batch and recursive methods. Batch SFM algorithms exist for static, clutter-free human-made scenes and small camera excursions. Many barriers remain, however, to achieve a real-time SFM solution for complex, large-scale scenes.

The issues of choice of representation and reliable extraction of features have been key issues in vision research [15]. Most SFM algorithms have employed points as features, extracted from images using techniques such as the SUSAN corner detector [20], with random sample consensus (RANSAC) used to determine the correspondence of points across image sequences. Using points as features, various batch SFM algorithms are summarized in Hartley and Zisserman [12] and Faugeras et al. [10]. For recursive SFM using points, the current state-of-the-art is perhaps best represented by Chiuso et al. [6] and McLauchlan [16]. Chiuso et al. have developed a real-time SFM system that can track large numbers of scene points and produce structure and motion estimates despite the presence of occlusion [6].

The Visual Computer (2003) 19:1–14
Digital Object Identifier (DOI) 10.1007/s00371-003-0205-3

McLauchlan has developed a hybrid batch/recursive technique for SFM using bundle adjustment, termed the variable-state dimension filter (VSDF) [16].

Our approach is similar in some ways to Chiuso et al. [6], but different in several key respects. One primary difference is that we use VPs and 3-D lines as features, rather than points. This greatly reduces the amount of processing required, and allows a conservative approach to initializing features into the map, and matching measurements to features. A nice feature of VPs is that they are invariant to translation, hence enabling the estimation of rotational errors to be decoupled from the estimation of translational errors.

Batch SFM using line features with manual correspondence has been addressed by Taylor and Kriegman [21]. More recent work by Taylor has integrated appearance-based methods with SFM to create a complete interactive 3-D reconstruction system [22]. In this paper, we consider recursive SFM using lines and fully automatic correspondence.

An automatic, recursive SFM algorithm must make data-association decisions such as 'which measurements correspond to previously mapped features?', 'which measurements correspond to new features?', and 'which measurements are spurious?'. The brittleness of most feature-detection methods is well known [14]; erroneous decisions about the origins of measurements can have disastrous consequences. Correspondence errors will lead to filter divergence. This motivates the development of methods that can make delayed decisions, assessing the consistency of data points obtained from multiple vantage points.

Batch algorithms for SFM often use techniques from robust statistics, such as RANSAC, to assess consistency across multiple images in a video sequence, but such a capability is harder to achieve in a recursive SFM implementation. In Chiuso et al.'s work, tracked features are not included in the full SFM solution until there is a high certainty that they provide good-quality data. This paper adopts a similar philosophy, using a technique to delay decisions about the origins of measurements and to perform consistent initialization of 3-D line features using data from multiple vantage points.

Our approach assumes that the scene contains sets of stationary parallel lines. This is a valid assumption in many human-made environments such as indoor offices and hallways, and outdoor urban environments. The technique assumes that calibration information for the omni-cam is available. While there has been a considerable effort in the use of omni-directional imagery in computer vision, we are unaware of any published work using omni-directional images to provide a full six-degrees-of-freedom, geometric SFM solution, using VPs and 3-D lines as features.

The structure of this paper is as follows: Sect. 2 describes the stochastic estimation framework. Section 3 describes the representation, detection, and estimation of vanishing points. Section 4 describes our dynamic programming approach to tracking parallel line segments in an image sequence. Section 5 describes the representation, initialization, and estimation of 3-D lines. Section 6 presents experimental results for the system for three different experiments, two performed indoors and one performed outdoors. Results for one of the indoor sequences are compared with a reference trajectory and a two-dimensional (2-D) map generated from a commercial laser/gyro navigation system. Finally, Sect. 7 summarizes the paper, contrasts it with other approaches to visual SFM, and suggests directions for future research.

2 Estimation framework

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 ease the data-association problem.

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) [5, 17, 19, 24], using laser range scanners [5, 9, 11, 17]. Some vision researchers have pursued similar approaches for small scenes and short camera excursions [8, 16].

This paper does not address the problems of large loop-closing and global relocalization. In related re-

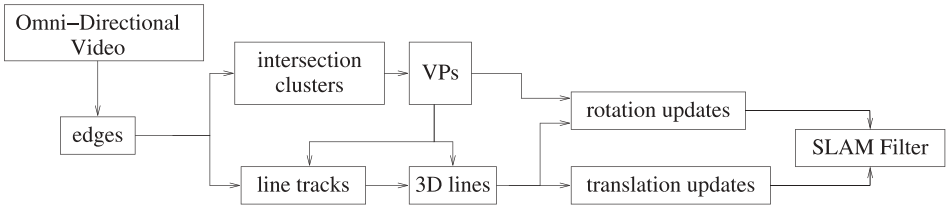


Fig. 1. Data flow graph

search, we have developed a method for large-scale mapping, called *Atlas*, and demonstrated large loop-closing and global relocalization for 2-D SLAM with laser and/or sonar data [4].

In this paper, we adopt a variable-dimension state-estimation framework, as employed in SLAM [19]. Figure 1 [TS a] summarizes the data flow in our system. Given a sequence of omni-directional images and detected linear features, our task is to estimate the position and orientation of 3-D scene landmarks (VPs and 3-D lines), and the pose of the robot (or camera on the robot) as each image was acquired. We join uncertain landmark states and robot poses in a common state vector, with a corresponding error covariance matrix:

$$\boldsymbol{x}_{\text{world}} := \begin{bmatrix} \boldsymbol{x}_{\text{robot}} \\ \boldsymbol{x}_{\text{map}} \end{bmatrix}, \tag{1}$$

$$\boldsymbol{P}_{\text{world}} := \begin{bmatrix} \boldsymbol{P}_{\text{rr}} & \boldsymbol{P}_{\text{rm}} \\ \boldsymbol{P}_{\text{mr}} & \boldsymbol{P}_{\text{mm}} \end{bmatrix}. \tag{2}$$

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

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{z}), \tag{3}$$

$$\boldsymbol{x} \leftarrow \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}, \tag{4}$$

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

where $g(\cdot)$, $\boldsymbol{G}_x$, and $\boldsymbol{G}_z$ are the mapping function and its Jacobians, and $\boldsymbol{P}_{zz}$ is the measurement covariance. Reobservation of a landmark improves the estimate of the observed landmark, the camera pose, 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 [13], 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.

3 Estimating vanishing points

Vanishing points (VPs) are the common directions of parallel 3-D lines [1]. In a perspective view, the VP is at the intersection of the images of the lines. We detect potential VPs in each view by finding clusters of line intersections. Subsequently, we use a random-sample consensus (RANSAC)-based method to seed initial clusters for an EM-based (expectation-maximization) refinement [2]. 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 up-

[TS a] Please check all figures carefully and supply new electronic version of better quality/resolution, if necessary.

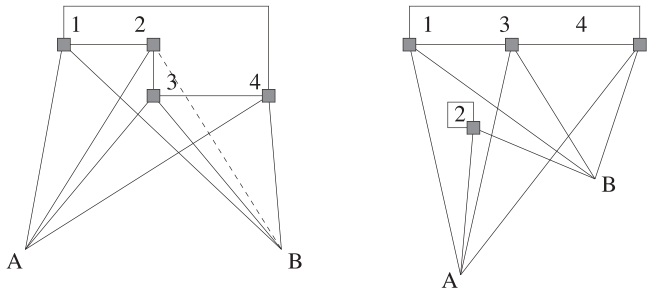


Fig. 2. The order of the objects cannot change unless they are on different surfaces and the motion is large

dated with all past views of the VP using the retained brief history of saved camera pose states.

We represent the VP with a 3-D unit vector in world coordinates. Since the VP is constrained to have unit length, it has only two effective degrees of freedom. This means that the uncertainty in the parameters of the VP 'live' on the constraint surface of the unit sphere. To represent this uncertainty in the covariance matrix of the EKF, we linearize the constraint at the VP to be the tangent plane of the unit sphere with its normal equal to the VP. Subsequently, we represent the plane's coordinate system by two basis vectors and project the covariance of the VP onto the plane.

The orthonormal basis vectors for the plane that is normal to the VP $\boldsymbol{v}_z$ are determined with the help of a constant vector $\boldsymbol{p}$ that is initialized to be orthogonal to the initial estimate of the VP:

$$\boldsymbol{p} = \begin{cases} n([1\,0\,0]^T \times \boldsymbol{v}_z) & \text{if } \boldsymbol{v}_z \cong [0\,0\,1]^T, \\ n([0\,0\,1]^T \times \boldsymbol{v}_z) & \text{otherwise,} \end{cases} \tag{6}$$

$$\boldsymbol{v}_x = n(\boldsymbol{v}_z \times \boldsymbol{p}), \tag{7}$$

$$\boldsymbol{v}_y = n(\boldsymbol{v}_z \times \boldsymbol{v}_x), \tag{8}$$

where the function $n(\boldsymbol{v})$ generates a unit vector in the same direction as $\boldsymbol{v}$: $n(\boldsymbol{v}) := \boldsymbol{v}/\|\boldsymbol{v}\|$.

We use the 2×3 matrix

$$\boldsymbol{T}_{\boldsymbol{v}} := [\boldsymbol{v}_x\ \boldsymbol{v}_y]^T \tag{9}$$

to project 3-D coordinates into the plane normal to the VP.

4 Tracking image lines

Lines are tracked across consecutive image frames using stochastic nearest-neighbor gating [3] augmented by a novel ordering constraint. When parallel lines are on a single surface, the relative order of the lines is maintained from subsequent views. Some lines may be occluded by convexities, but their order can change only if the lines are on different surfaces and the motion is large with respect to the size of the surfaces (Fig. 2). Since in practice this is rare, we can exploit this constraint to aid in tracking lines whose prediction gates overlap. Nearest-neighbor tracking is not robust enough because many typical scenes have groups of lines that are closer together than the tracking prediction uncertainty.

The lines from the previous view are projected into the current view and then are matched to the newly extracted image lines. By sorting the lines around each corresponding VP we can apply a modified version of the standard longest common substring algorithm, which is efficiently solved with dynamic programming [7], to find the best match while maintaining the ordering constraint. If we imagine the VP as the north pole of a globe (Fig. 3a), then all the line segments run north–south, and they can be lexicographically sorted by the longitude and latitude of their midpoints. The mean color to the left and right of a line are used to further reduce the chance of false matches.

5 Mapping 3-D lines

In addition to VPs, the system tracks local, parallel 3-D line segments that share a common VP. These segments have six degrees of freedom (DOF), 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 locations of the endpoints along the line. The endpoints are maintained

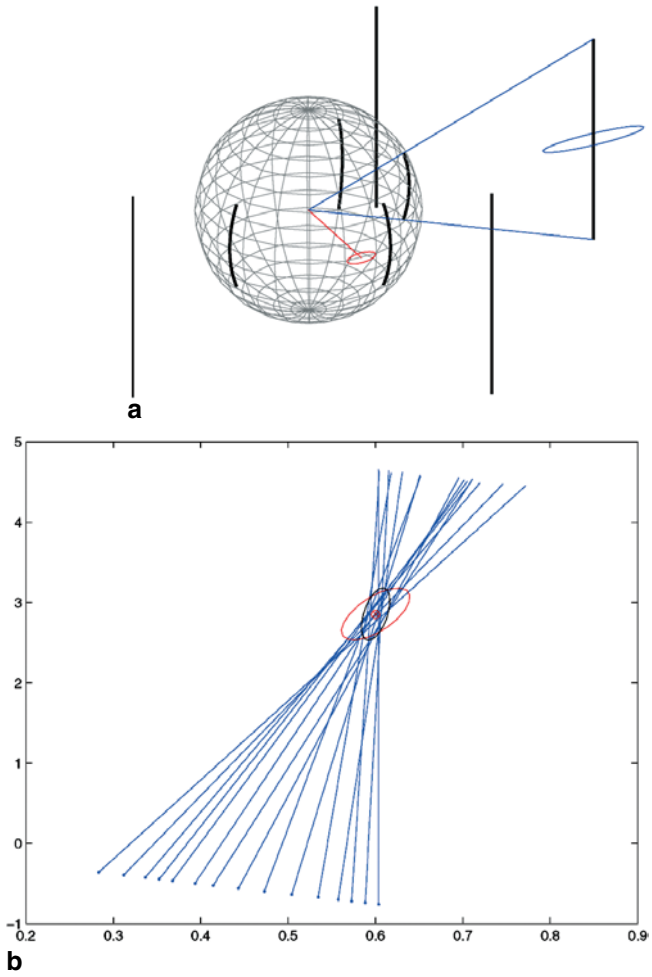


Fig. 3. a Projection of a 3-D line. **b** View in normal plane of initialization of a 3-D line

as extra information outside the EKF state vector because of the non-Gaussian nature of occlusions. Thus only two unique state elements are maintained for each 3-D line landmark in the EKF. This offset of the 3-D line is parameterized as a point in the plane which passes through the origin and is orthogonal to the corresponding VP of the line. This plane is parameterized by the same basis vectors as in (9).

Figure 3 shows the projection of a 3-D line and its covariance onto the unit sphere. The system updates 3-D lines by projecting them into the current view, and comparing them with the corresponding line extracted from the current view.

Due to the need to maintain estimates of correlations between all features in the map, the computational complexity of stochastic mapping is $\mathcal{O}(n^2)$, where n is the number of features mapped [18]. The current implementation simply drops older features when the number of features in the state vector exceeds a preset constant limit (fifty 3-D line segments in the experiments reported below).

6 Experimental results

We present experimental results using three omni-directional video sequences. The experimental system consists of a digital video camcorder attached to a parabolic mirror which was oriented vertically

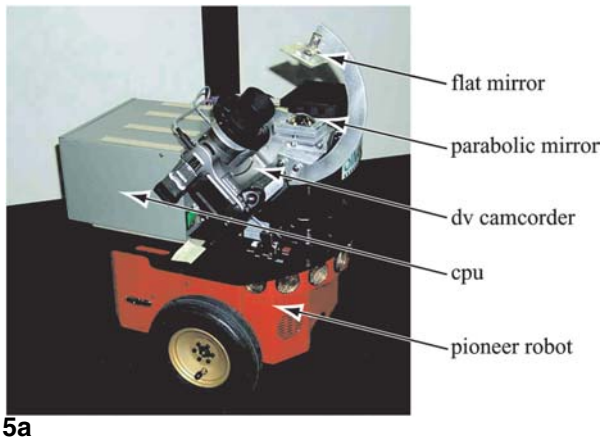
```
for each image
  extract edge contours from Laplacian of Gaussian
  find mirror calibration marks and calibrate projection function
  map edge contour pixel locations to rays
  fit lines to contours and compute line covariances
  find initial intersection clusters with RANSAC
  refine intersection clusters with EM to get current view's VPs
  label lines with best VP from current view
  propagate KF to current time
  project KF VPs and preliminary VP tracks into current view
  match projected VPs with current VPs
  for each matched KF VP
    update KF state and covariance
  for each preliminary VP
    extend preliminary VP track
  for each unmatched VP
    initialize new VP track
  for each preliminary VP track longer than N frames
    check if VP is invariant to translation
    map new VP into KF state and covariance using current view
    update new VP with past views
  for each VP in KF
    project all corresponding lines and preliminary lines into current view
    sort projected lines and current view lines around VP
    match lines using dynamic programming to enforce order constraint
  for each match KF line
    compute normalized residual
    if residual < outlier threshold
  update KF state and covariance
  for each matched preliminary line
    extend line track
  for each unmatched line
    initialize new track
  for each preliminary line track longer than N frames
    check to see if intersection is consistent with KF covariance
    if consistent line intersection
  initialize KF 3-D line and covariance from first and last views
  update 3-D line from remaining past views
    if more than M lines in KF
  find line with oldest reobservation time
  delete line and corresponding covariance elements
  delete oldest robot pose and covariance
  duplicate current robot pose and covariance
  delete preliminary VP and line tracks older than N frames
```

Fig. 4. Summary of the recursive SFM algorithm using VPs and 3-D lines

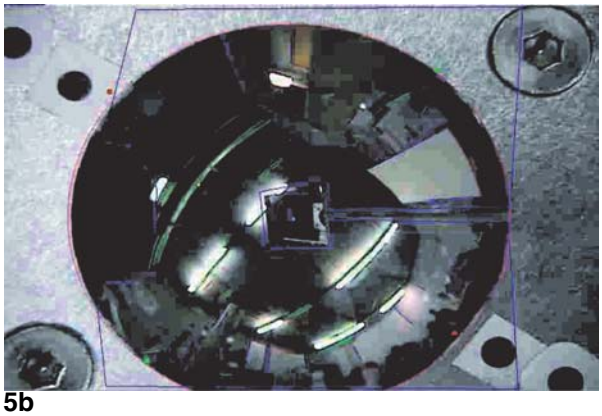
for the indoor sequences and horizontally for the outdoor sequence.

6.1 Indoor sequence A: mobile-robot experiment

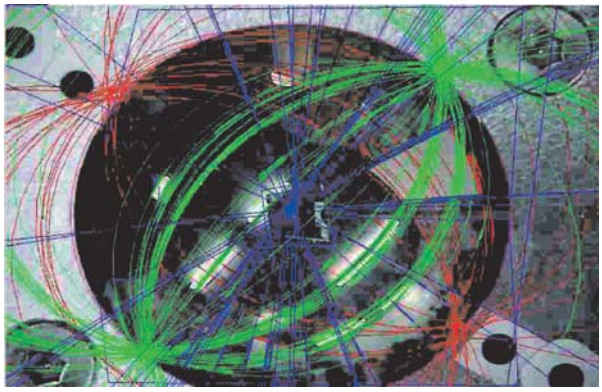
For the first indoor sequence, the camera was mounted on a small mobile robot as shown in Fig. 5a. To maintain the calibration of the omni-camera despite vibration during data acquisition, fiducial marks adjacent to the mirror (visible in Fig. 5) were registered. For synchronization, we used a software modem to encode time stamps from the robot in the camcorder's audio channel. The current implementation of the algorithm uses MATLAB and has not been optimized for speed. The processing time was roughly one to three frames per second. We are developing an optimized C++ implementation to enable real-time operation.



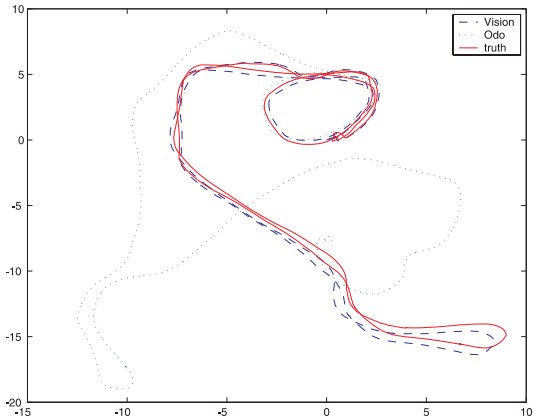
5a



5b



5c



6

Fig. 5. a The omni-directional video camera mounted on a Pioneer robot. **b** Edges and VPs extracted. **c** Edges extended to infinite lines

Fig. 6. Comparison of trajectories estimated from the SFM algorithm (*dashed line*), a commercial laser/gyro navigation system (*solid line*), and odometry for the indoor experiment (*dotted line*). The total path length was 106 m and the robot returned to within approximately 30 cm of its starting point

The video sequence consisted of 2400 frames with a trajectory length of 106 m. Odometry data from the robot was used for EKF state projection. The error drift rate was approximately 15 degrees per minute. The robot moved at a speed of approximately 25 cm per second. Sample images and features are shown in Fig. 5b,c. Laser scanner data and commercial robot-navigation software provided a ground-truth camera pose with an accuracy of approximately 5 cm and 0.5 degrees.

Figure 6 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. Compar-

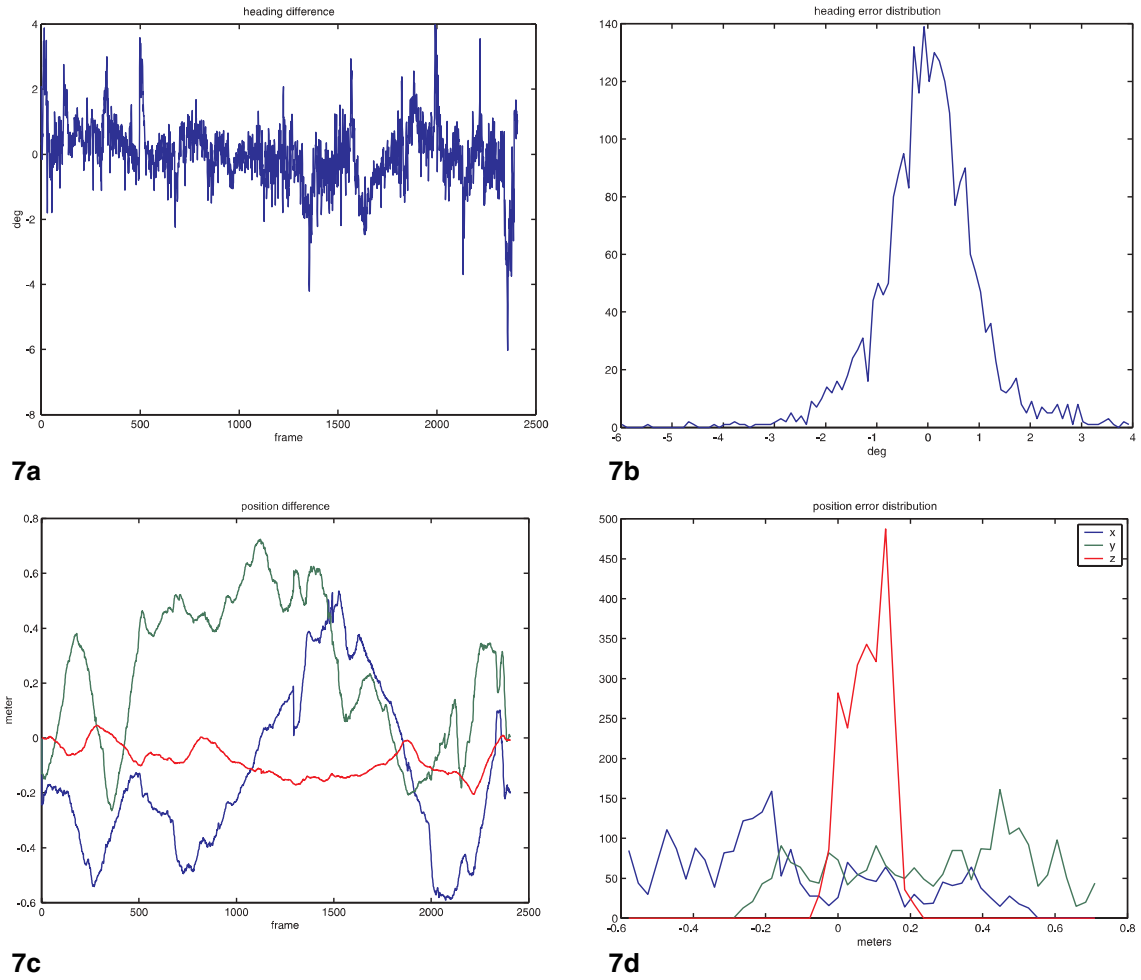
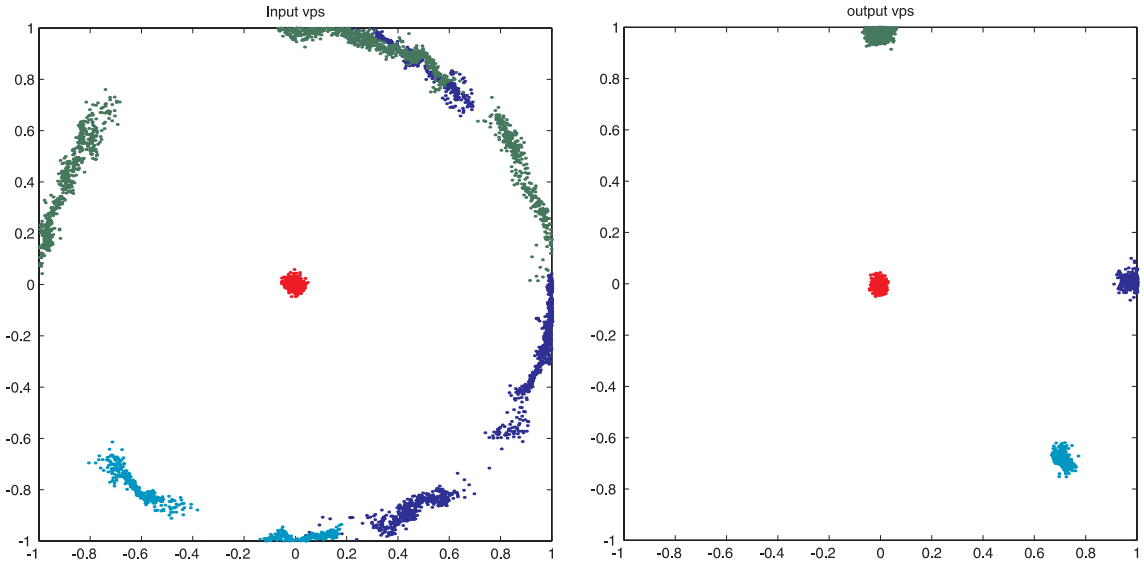


Fig. 7. State-estimation errors. **a** Difference between heading estimated by the laser/gyro system and the SFM algorithm. **b** Histogram of heading errors (note that there is only a 1 degree standard deviation). **c** Difference between translation (x, y, z) estimated by the laser/gyro system and the SFM algorithm. **d** Histogram of translation errors

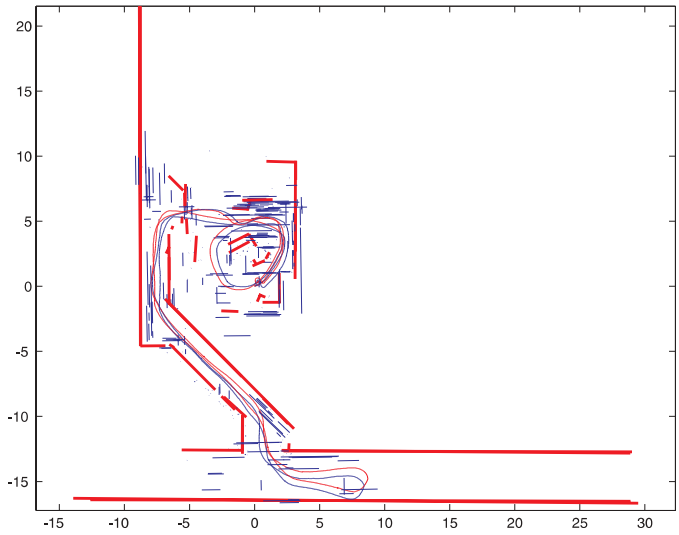
isons with ground-truth are possible only about one rotation axis (yaw). However, our algorithm computes a full six-DOF solution for the camera trajectory. Figure 7 shows the distribution of rotational and translation errors of the visual SFM output as compared to the laser/gyro estimates. The standard deviation of heading errors was one degree, and the standard deviations for translation errors were 30 cm in x and y and 6 cm in z. Figure 8 shows the distribution of vanishing point directions referenced to odometry and to the visual SFM output. This demonstrates the method's ability to decouple estimation of rotational and translational errors. Finally, Fig. 9 shows two views of the map estimated by the algorithm, compared with a 2-D reference laser map.

6.2 *Indoor sequence B: body-mounted data acquisition*

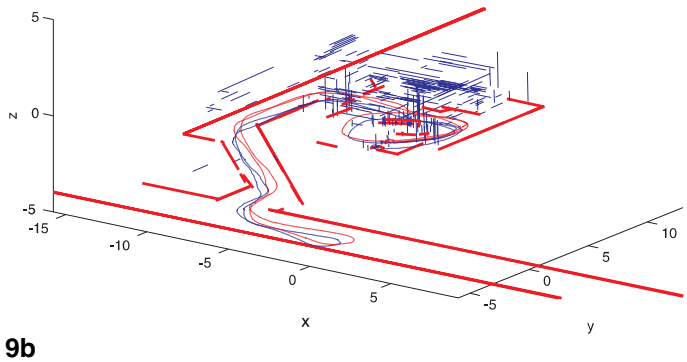
The omni-camera was mounted on a novel 'backpack' data-acquisition system that also contained an inertial measurement unit and a radar Doppler velocity sensor. The system was used to acquire a video sequence for a person walking around an indoor environment. Figure 10 shows examples of images from



8a 8b



9a



9b

Fig. 8. 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 robot orientations estimated by our algorithm

Fig. 9. 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

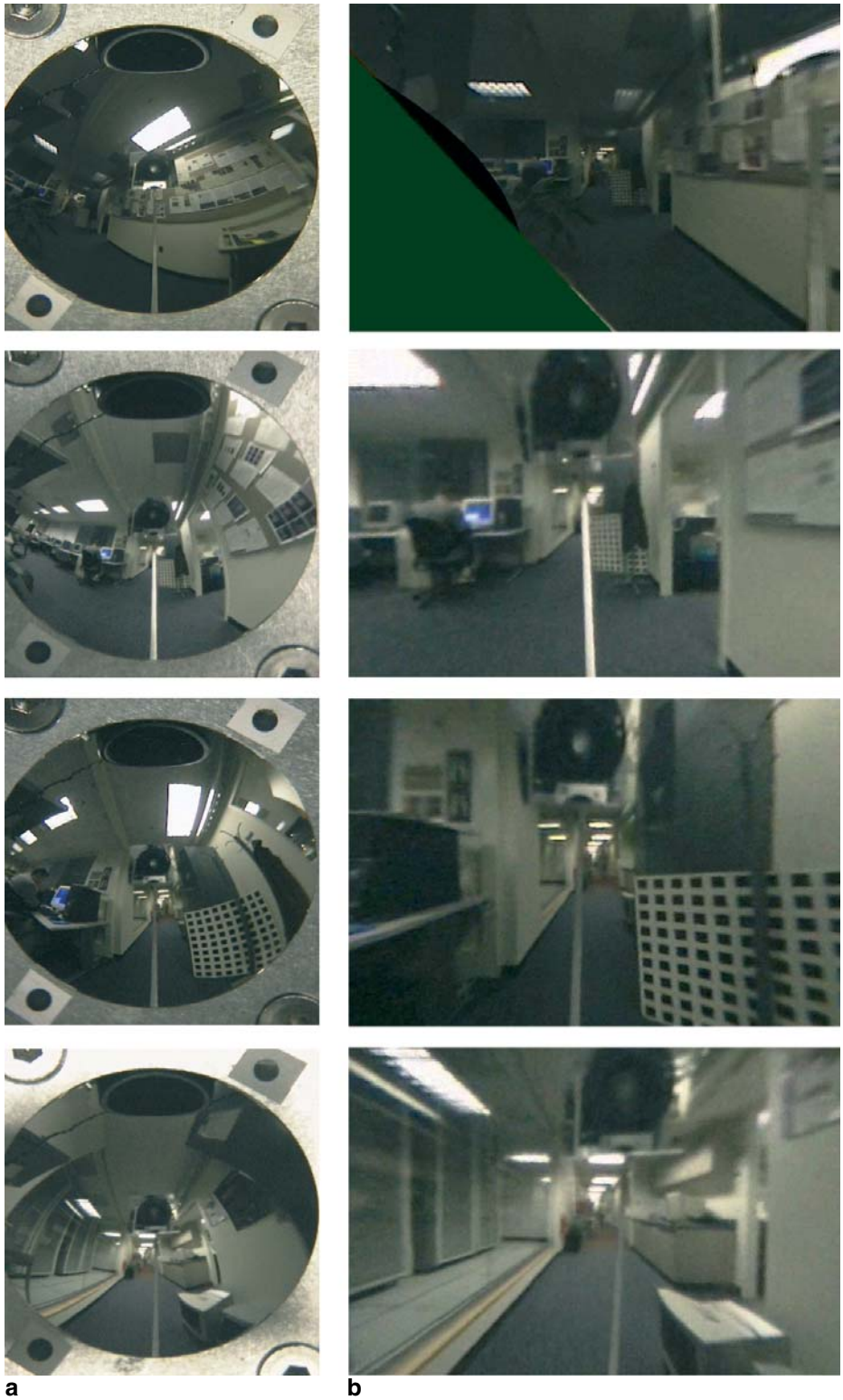
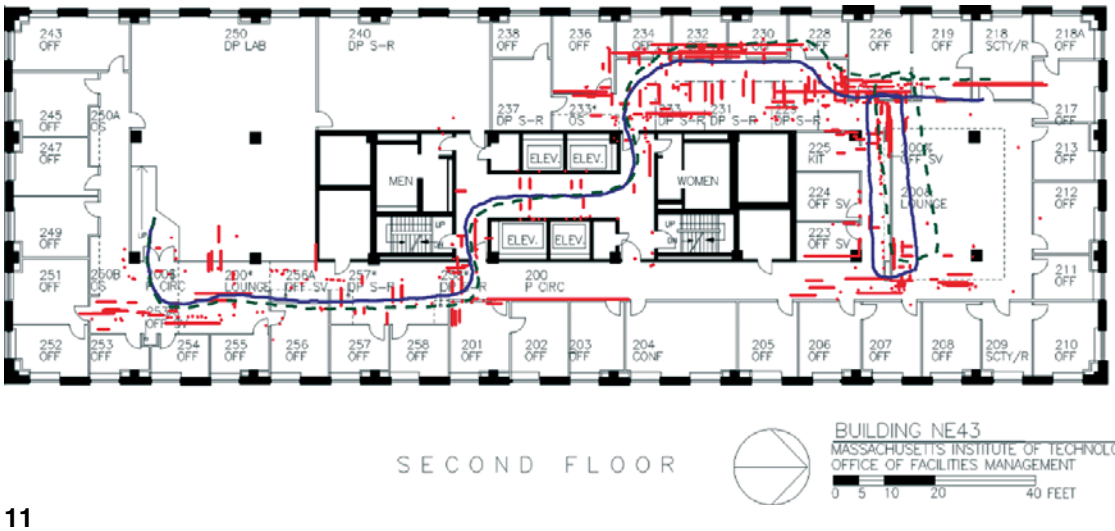
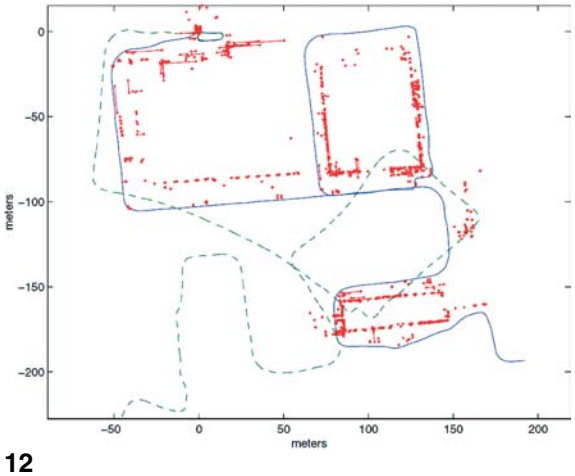


Fig. 10. An example application of accurate per-frame orientation estimation. In column (a) we show four frames of the input video and in column (b) we depict the results of rendering a new view of each frame with the virtual camera's rotation constrained to align with the world's x axis



11



12

Fig. 11. Overlay of estimated camera path and scene structure with an architectural drawing for the body-mounted indoor experiment

Fig. 12. Comparison of trajectories estimated from the SFM algorithm and odometry for the outdoor experiment. The image sequence is 17 000 frames acquired over a 946-m path

the sequence after stabilization based on VP estimation. Figure 11 shows the SFM output and estimated camera trajectory superimposed on a floor plan of the environment (the 2nd floor of MIT's Laboratory for Computer Science).

6.3 Outdoor experimental results

The omni-camera was mounted on the Argus city-scanning rolling platform [23] and used to obtain an outdoor video sequence consisting of 17 000 frames with a path length of 946 m. Figure 12 shows the estimated map and camera trajectory, compared to odometry. Although no ground-truth is available for this sequence, we have overlaid the path and structure on top of a site plan for a qualitative evaluation, as shown in Fig. 13.

The experiment terminated when the robot entered a featureless region in which no vanishing points were visible. Recovery from such situations is possible using the *Atlas* framework for large-scale mapping [4].

7 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

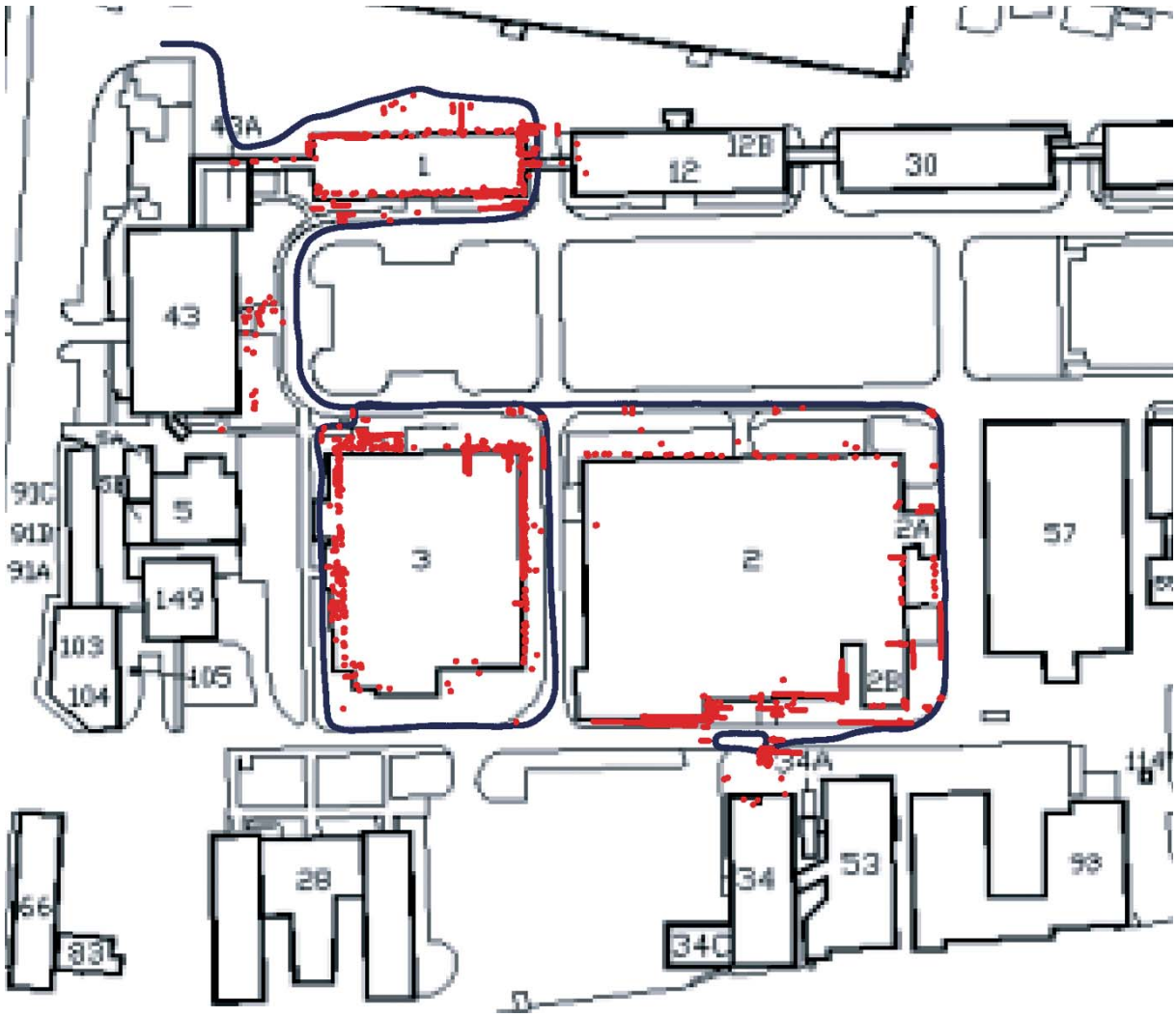


Fig. 13. Estimated camera path and 3-D line segments overlaid on site plan for outdoor experiment

a recursive state-estimation framework [19]. The approach has been demonstrated with off-line processing of both indoor and outdoor image sequences. In the mobile-robot indoor experiment (Sect. 6.1), 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. In both the body-mounted experiment (Sect. 6.2) and the outdoor experiment (Sect. 6.3), there is excellent agreement between the estimated scene structure and available architectural drawings.

In ongoing research, the methods described in this paper are being combined with the *Atlas* framework for scalable mapping [4] to enable computationally efficient large-scale mapping and loop-closing. In addition, methods are being developed for texture estimation and aggregate feature matching.

Acknowledgements. This research has been funded in part by NSF Career Award BES-9733040, the MIT Sea Grant College Program under Grant No. NA86RG0074 (Project No. RCM-3), the Office of Naval Research under Grant No. N00014-97-0202, and by Draper Laboratories under Contract Nos. DL-H-516617, DL-H-526716, and DL-H-539054.

References

1. Antone M, Teller S (2000) Automatic recovery of relative camera rotations for urban scenes. In: International Conference on Computer Vision and Pattern Recognition, Vol II, pp 282–289
2. Antone ME (2001) Robust camera pose recovery using stochastic geometry. PhD thesis, Massachusetts Institute of Technology
3. Bar-Shalom Y, Fortmann TE (1988) Tracking and data association. Academic Press Limited, San Diego, CA, 1988
4. Bosse M, Newman P, Leonard J, Teller S (2002) An atlas framework for scalable mapping. Marine Robotics Laboratory Technical Memorandum 02-04, Massachusetts Institute of Technology
5. Castellanos JA, Montiel JMM, Neira J, Tardos JD (1999) The SPmap: a probabilistic framework for simultaneous localization and map building. IEEE Trans Robot Autom 15(5):948–952
6. Chiuso A, Favaro P, Jin H, Soatto S (2000) 3-d motion and structure from 2-d motion causually integrated over time: implementation. In: Sixth European Conference on Computer Vision, pp[TS b]
7. Cormen T, Leiserson C, Rivest R (2001) Introduction to algorithms. MIT Press[TS c]
8. Davison AJ (1998) Mobile robot navigation using active vision. PhD thesis, University of Oxford
9. Dissanayake MWMG, Newman P, Durrant-Whyte HF, Clark S, Csorba M (1999) An experimental and theoretical investigation into simultaneous localization and map building. In: Sixth International Symposium on Experimental Robotics, pp 265–274
10. Faugeras O, Luong Q-T, Papadopoulo T (2001) The geometry of multiple images. MIT Press[TS b]
11. Guivant J, Nebot E (2001) Optimization of the simultaneous localization and map building algorithm for real time implementation. IEEE Trans Robot Autom 17(3):242–257
12. Hartley RI, Zisserman A (2001) Multiple view geometry in computer vision. Cambridge University Press [ISBN: 0521623049]
13. Leonard JJ, Rikoski RJ, Newman PM, Bosse M (2002) Mapping partially observable features from multiple uncertain vantage points. Int J Robot Res, to appear[TS d]
14. Lozano-Pérez T (1989) Foreword. In: Cox IJ, Wilfong GT (eds.) Autonomous robot vehicles. Springer, Berlin
15. Marr D (1982) Vision. Freeman, New York
16. McLauchlan PF (2000) A batch/recursive algorithm for 3d scene reconstruction. In: International Conference on Computer Vision and Pattern Recognition, Hilton Head, SC, Vol 2, pp 738–743
17. Moutarlier P, Chatila R (1989) An experimental system for incremental environment modeling by an autonomous mobile robot. In: 1st International Symposium on Experimental Robotics, Montreal
18. Moutarlier P, Chatila R (1989) Stochastic multi-sensory data fusion for mobile robot location and environment modeling. In: 5th International Symposium on Robotics Research, pp 207–216
19. Smith R, Self M, Cheeseman P (1987) A stochastic map for uncertain spatial relationships. In: 4th International Symposium on Robotics Research, pp[TS b]
20. Smith SM, Brady JM (1997) Susan – a new approach to low level image processing. Int J Comput Vision 23(1):45–78
21. Taylor CJ, Kriegman DJ (1995) Structure and motion from line segments in multiple images. IEEE Trans Pattern Anal Mach Intell 17(11):1021–1032
22. Taylor CJ (2000) Videoplus: a method for capturing the structure and appearance of immersive environments. In: Second Workshop on 3D Structure from Multiple Images of Large-scale Environments, pp 187–204
23. Teller S, Antone M, Bodnar Z, Bosse M, Coorg S (2001) Calibrated, registered images of an extended urban area. In: International Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, Vol 1, pp 813–820
24. Thrun S (2001) An online mapping algorithm for teams of mobile robots. Int J Robot Res 20(5):335–363
25. Triggs B, Zisserman A, Szeliski R (eds.) (1999) Vision algorithms, theory and practice: International Workshop on Vision. Springer, Berlin

Photographs of the authors and their biographies are given on the next page.

[TS b] Please supply the page numbers, if possible.
[TS c] Is this a book? Please clearify.
[TS d] Please update if possible.

MICHAEL BOSSE is a PhD candidate in the Computer Graphics Group at LCS, where he has been developing scalable navigation systems using sonar, laser, or omnidirectional video. He earned his BS and MS in 1997 from Boston University, where he built a vision-aided navigation system for an autonomous helicopter. His research interests include computer vision, robotics, and navigation.

RICHARD J. RIKOSKI is a doctoral candidate in the Marine Robotics Laboratory at MIT. His research interests are marine robotics and ocean AI. His research area is incoherent sonar perception for dynamic, imprecisely navigating undersea robots. He holds the degrees of BS in Mechanical Engineering and Economics from Carnegie Mellon University (1998) and SM in Ocean Engineering from MIT (2001).

JOHN J. LEONARD is Associate Professor of Ocean Engineering at MIT. His research addresses the problems of autonomous navigation and mapping for autonomous mobile robots, with emphasis on the problems of sonar perception and navigation for autonomous underwater vehicles. He holds the degrees of BSE in Electrical Engineering and Science from the University of Pennsylvania (1987) and DPhil in Engineering Science from the University of Oxford (1994). He has been a member of the MIT faculty since 1996. From 1991 to 1996, Prof. Leonard was a Postdoctoral Fellow and Research Engineer at the MIT Sea Grant College Program, where he was part of a team that developed the Odyssey II autonomous underwater vehicle. He is an associate editor of the IEEE Journal of Oceanic Engineering and the IEEE Transactions on Robotics and Automation.

SETH TELLER obtained a PhD from UC Berkeley in 1992, focusing on accelerated rendering of complex architectural environments. After postdoctoral research at the Computer Science Institute of the Hebrew University of Jerusalem Institute of Computer Science, and Princeton University's Computer Science Department, he joined MIT's Electrical Engineering and Computer Sciences Department, and MIT's Lab for Computer Science in 1994. Prof. Teller now leads the Computer Graphics Group in LCS, where he and his students pursue innovative strategies for architectural environment capture and simulation, and the design of pervasive computing infrastructure, algorithms, and applications.