

Automated Camera Calibration and 3D Egomotion Estimation for Augmented Reality Applications

Dieter Koller^{1,2,3}, Gudrun Klinker¹, Eric Rose¹, David Breen⁴, Ross Whitaker⁵,
and Mihran Tuceryan⁶

¹ Fraunhofer Project Group for AR at ZGDV, Arabellastr.17 (at ECRC), 81925 Munich, Germany

² EE Dept., California Inst. of Technology, MC 136-93, Pasadena, CA 91125

³ Autodesk, Inc., 2465 Latham St., Suite 101, Mountain View, CA 94040

⁴ Computer Graphics Lab., California Inst. of Technology, MC 348-74, Pasadena, CA 91125

⁵ EE Dept., 330 Ferris Hall, U. of Tennessee, Knoxville, TN 37996-2100

⁶ Dept of Comp & Info Science, IUPUI, 723 W. Michigan St, Indianapolis, IN 46202-5132

Email: dieter.koller@autodesk.com

Abstract. This paper addresses the problem of accurately tracking the 3D motion of a monocular camera in a known 3D environment and dynamically estimating the 3D camera location. For that purpose we propose a fully automated landmark-based camera calibration method and initialize a motion estimator, which employs extended Kalman filter techniques to track landmarks and to estimate the camera location at any given time. The implementation of our approach has been proven to be efficient and robust and our system successfully tracks in real-time at approximately 10 Hz. We show tracking results of various augmented reality scenarios.

1 Introduction

Augmented reality (AR) is a technology in which a user's view of the *real* world is enhanced or augmented with additional information generated by a computer. The enhancement may consist of rendered virtual geometric objects placed into the environment, or a display of non-geometric information about existing real objects. Using AR technology, users can interact with a mixed virtual and real world in a natural way.

This paradigm for user interaction and information visualization provides a promising new technology for many applications. AR is being explored within a variety of scenarios. The most active application area is medicine, where AR is used to assist surgical procedures by aligning and merging medical images into video [1; 2; 3; 4]. For manufacturing AR is being used to direct workers wiring an airplane [5]. In telerobotics AR provides additional spatial information to the robot operator [6]. AR may also be used to enhance the lighting of an architectural scene [7], as well as, provide part information to a mechanic repairing an engine [8]. For interior design AR may be used to arrange virtual furniture in a real room [9].

1.1 Technical Contribution

A video-based AR system can be regarded as having two cameras: a real one generating live video of the real environment, and a virtual one producing 3D graphics to be merged with the live video stream. Both cameras must have the same internal and external parameters in order for the real and virtual objects to be properly aligned. This is accomplished by an initial calibration of the real camera and a dynamic update of its external parameters. From the vision point of view, this is one of the most challenging technical problems that needs to be addressed in order to produce a useful and convincing video-based augmented reality system ([10]).

More traditional augmented reality approaches employ magnetic tracking devices for sensing position and orientation of a moving camera (e.g. [8; 11]). They suffer, however, from (a) limited range (3-5m), (b) interference with ferromagnetic objects of the environment, and (c) lack of portability.

We therefore focused primarily on vision-based algorithms and decided on using landmark-based calibration and tracking to make it more tractable. We are using corners of rectangular patterns attached to a wall as landmarks and track them using extended Kalman filter techniques based on an acceleration-free constant angular velocity and constant linear acceleration motion model. We demonstrate the robustness and accuracy of our tracker within various augmented reality application.

1.2 Related Work

A number of groups have explored the topic of camera tracking for augmented reality. Some researcher [12; 13] have argued that a simple view based, calibration free approach for real-time visual object overlay is sufficient. This is definitely true for certain applications, where no direct metric information is necessary. *Interactive AR*, however, requires the more complex pose calculation based approach, which allows the decomposition of the image transformation into camera/object pose and the full perspective projection in order to calculate 3D collision detection and interaction between real and virtual objects, as in [14].

A similar argument is being used in [15] in the context of enhanced reality in medicine [4], where near real-time calibration is performed for each frame based on a few fiducial marks. In [11; 16] a hybrid vision and magnetic system is used to improve the accuracy of tracking a camera over a wide range of motions and conditions.

Tracking known objects in 3D space and ego-motion estimation (camera tracking) have a long history in computer vision (e.g. [17; 18; 19; 20]). Constrained 3D motion estimation is being applied in various robotics and navigation tasks. Much research has been devoted to estimating 3D motion from optical flow fields (e.g. [21]) as well as from discrete moving image features like corners or line segments (e.g. [22; 23; 24]), often coupled with structure-from-motion estimation, or using more than two frames (e.g. [25]). The theoretical problems seem to be well understood, but robust implementation is difficult. The development of our tracking approach and the motion model has mainly been influenced by the one described in [20].

1.3 Outline of the Paper

We start by describing our automated camera calibration procedure in Section 2. In Section 3 we explain the motion model employed by our Kalman filter. The Kalman filter based tracking procedure is then outlined in Section 4. We finally present some results in Section 5 and close with a conclusion in Section 6.

2 Automated Camera Calibration

We propose an automated calibration procedure which determines internal parameters (focal length and focal center according to the standard pinhole camera model) and external camera parameters (camera pose). In the current implementation the internal parameters are fixed during a session and only the external parameters are estimated.

A highly precise camera calibration is required for a good initialization of the tracker. For that purpose we propose a two step calibration procedure, in which we attempt to find the image locations of landmarks placed in the 3D environment at known 3D locations (cf. Figure 3). This addresses the trade-off between high precision calibration and minimal or no user interaction. We use dark rectangular cardboard as landmarks. In the first step we perform an initial camera calibration based on the image location of the centroids of dark blobs (landmarks) which we extract in the image. This bootstraps the second step consisting of a constraint search for additional image features (corners); thus improving the calibration. We are using the camera calibration algorithm described in [26] and implemented in [27].

The next subsection describes our algorithm for finding dark image blobs. The constrained search for projected model squares is addressed in the context of acquiring measurements for the Kalman filter in Subsection 4.2.

2.1 Finding Dark Image Blobs

The algorithm for finding dark blobs in the image is based on a *watershed* transformation, a morphological operation which decomposes the whole image into connected regions (*puddles*) divided by watersheds (cf. [28]).

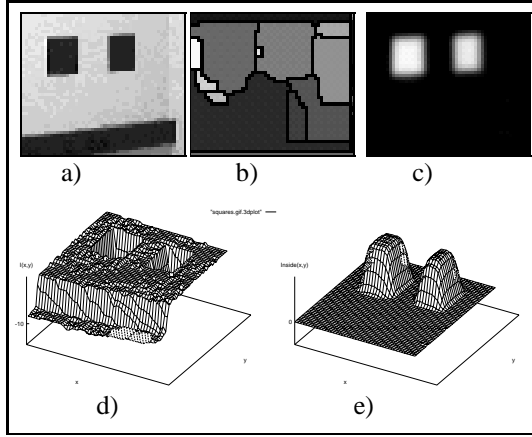


Fig. 1: (a) Subimage with dark squares, (b) watershed transformation with greycoded regions (watershed are drawn in black), (c) greyscale inside operation, measuring the depth of *puddles*, (d) and (e) show 3D plots of images (a) and (c), respectively.

identification numbers or with multiple assignments of the same number are discarded. Using this scheme, the tracker can calibrate itself even when some of the model squares are occluded or outside the current field of view (see Figure 6 a)).

3 Motion Model For Rigid Body Motion

Tracking can be stated as estimating the motion parameters according to a known motion model. Our application scenario suggests a fairly irregular camera and object motion within all 6 degrees of freedom¹. We assume an acceleration free motion of the camera due to the lack of *a priori* knowledge about the forces changing the motion. It is well known that in this case a general motion can be decomposed into a constant translational velocity \mathbf{v}_c of the objects centroid c , and a rotation with constant angular velocity $\boldsymbol{\omega}$ around an axis through the centroid (of the camera). (cf. Figure 2 and [29]). The motion equation of a camera point \mathbf{p} wrt. world coordinates is then given by: $\dot{\mathbf{p}} = \mathbf{v}_c + \boldsymbol{\omega} \times (\mathbf{p} - c)$. In this equation the constant rotation is wrt. the moving centroid and hence no motion invariant wrt. to the world coordinate frame. Instead, the center of rotation itself is moving with $c(t) = c(t_0) + \mathbf{v}_c(t - t_0)$. Substituting this into the motion equation produces:

$$\dot{\mathbf{p}}(t) = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{p} + \mathbf{a}t \quad (1)$$

with $\mathbf{v}(t_0) = \mathbf{v}_c - \boldsymbol{\omega} \times c(t_0)$ and $\mathbf{a} = -\boldsymbol{\omega} \times \mathbf{v}_c = \text{const.}$ The rotation is now with respect to world coordinates. However, an additional acceleration term \mathbf{a} is added. The integration yields (cf. [20; 30]):

$$\mathbf{p}(t + \Delta t) = R(\boldsymbol{\theta}) \mathbf{p} + S(\boldsymbol{\theta}) \mathbf{v} \Delta t + T(\boldsymbol{\theta}) \mathbf{a} \left(\frac{\Delta t}{2}\right)^2,$$

¹ In an AR application the camera can be hand held or even head mounted so the user is free to move the camera in any direction.

Using this transformation a dark blob surrounded by a bright area provides a strong filter response related to the depth of the *puddle* (cf. Fig. 1). The *deepest* and most compact blobs (puddles) are then matched against the known 3D squares. For this purpose, the squares contain one or more small red squares at known positions, representing binary encodings of the identification numbers of the model squares. The red squares are barely visible in the green and blue channels of the video camera. Thus we can apply a simple variant of a region growing algorithm to the green color channel to determine the borders of each black square. After fitting straight lines to the border, we sample each black square in the red color channel at the supposed locations of the internal red squares to obtain the bit pattern representing the model id. Blobs with invalid

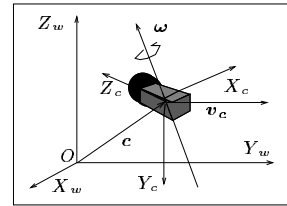


Fig. 2: The world (X_w, Y_w, Z_w) and camera (X_c, Y_c, Z_c) coordinate frames, and translational (\mathbf{v}_c) and rotational ($\boldsymbol{\omega}$) velocities.

with

$$R(\theta) = \mathbf{I}_3 + \frac{\sin \theta}{\theta} \Theta + \frac{1 - \cos \theta}{\theta^2} \Theta^2 = e^\Theta, \quad S(\theta) = \mathbf{I}_3 + \frac{1 - \cos \theta}{\theta^2} \Theta + \frac{\theta - \sin \theta}{\theta^3} \Theta^2$$

$$T(\theta) = \mathbf{I}_3 + 2 \frac{\theta - \sin \theta}{\theta^3} \Theta + \frac{\theta^2 - 2(1 - \cos \theta)}{\theta^4} \Theta^2$$

and $\theta = \omega \Delta t = (\theta_x, \theta_y, \theta_z)$, $\theta = \|\theta\|$. Θ is the skew-symmetric matrix to vector θ :

4 Tracking and Egomotion Estimation

With tracking our system is able to cope with dynamic scene changes and camera motions. Our tracking approach currently uses the corners of squares attached to moving objects or walls (cf. Figure 3), which have already been used for camera calibration.

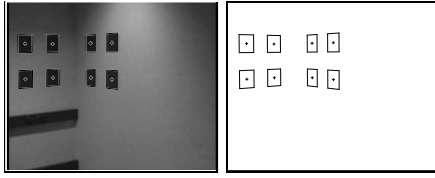


Fig. 3: Our vision-based tracking approach currently tracks the corners of squares. Left: image with eight squares. Right: detected squares only.

sition vector \mathbf{t} , the rotation vector ϕ , the translational and angular velocity \mathbf{v} and ω , respectively, and the translational acceleration \mathbf{a} : $\mathbf{s} = \{\mathbf{t}, \phi, \mathbf{v}, \omega, \mathbf{a}\}$.

The extended Kalman filter (EKF) equations can be found in most related textbooks, e.g., [31]. We just want to add an implementation note: the standard Kalman filter calculates the gain in conjunction with a recursive computation of the state covariance. This requires a matrix inversion of the dimension of the measurement vector, which can be large as in our application. However, the matrix inversion can be reduced to one of the state dimensions using the *information matrix* formalism. The *information filter* recursively calculates the inverse of the covariance matrix (= information matrix) (cf. [32]):

$$P_k^+{}^{-1} = P_k^-{}^{-1} + H_k^T R_k^{-1} H_k, \quad (2)$$

where P_k^+ denotes the updated covariance matrix, P_k^- the prediction, H_k the jacobian of the measurement function, and R_k the measurement noise matrix, each at time k . The update equation for the state $\hat{\mathbf{s}}_k^+$ then becomes:

$$\hat{\mathbf{s}}_k^+ = \hat{\mathbf{s}}_k^- + K_k(\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{s}}_k^-)) \quad \text{with} \quad K_k = \left(P_k^-{}^{-1} + H_k^T R_k^{-1} H_k \right)^{-1} H_k^T R_k^{-1}, \quad (3)$$

which requires the inverse of the updated covariance matrix P_k^+ of Eqn 2. Inverting R_k is straightforward since we assume independent measurements producing a diagonal measurement noise matrix R_k . The transition equation (prediction) can be found in [30].

4.2 Kalman Filter Measurements

Currently we use the image positions of corners of squares as measurements, i.e., our $8 \cdot n$ dimensional measurement vector \mathbf{z} comprises the x and y image positions of all of the vertices (corners) of the n squares we find in the image. A measurement \mathbf{z} is mapped to the state \mathbf{s} by means of the measurement function \mathbf{h} : $\mathbf{z} = \mathbf{h}(\mathbf{s})$.

The image corners are extracted in a multi-step procedure outlined below and in Figure 4. Assume that we are looking for the projection $p_i = l_j \cap l_k$ of the model vertex $v_i = m_j \cap m_k$ which is given by the intersection of the model lines m_j , and m_k (l_j and l_k are the image projections of the model lines m_j and m_k).

Once a complete camera calibration has been performed as described in Section 2, we can switch to the tracking phase, i.e., update the pose and motion parameters of the camera by keeping the internal camera parameter constant. We employ extended Kalman filter (EKF) techniques for optimal pose and motion estimation using the motion described in Eqn. 1.

4.1 Extended Kalman Filter

Our state vector \mathbf{s} of the Kalman filter comprises the following 15 components: the po-

- Predict image locations for model lines m_j and m_k .
- Subsample these predicted lines (e.g., into 5 to 10 sample points).
- Find the maximum gradient normal to the line at each of those sample points using a search distance given by the state covariance estimate. We use only 8 possible directions and extract the maximum gradient with sub-pixel accuracy.
- Fit a new line l_j to the extracted maximum gradient points corresponding to the predicted model line m_j .
- Find the final vertex $p_i = l_j \cap l_k$ by intersecting the correspondent lines l_j and l_k .

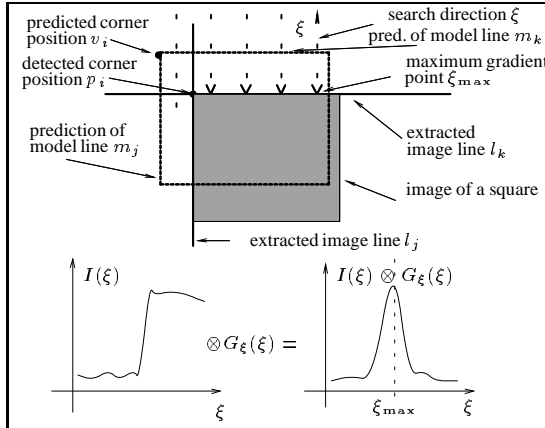


Fig. 4: We use corners as intersections of matched line segments as measurements. These line segments are fitted from maximum gradient points which are produced from a one dimensional convolution with a derivative of a gaussian kernel G_ξ normal to the projection of the image line (ξ is a parameterization normal to the line and I is the image intensity).

This procedure allows us to obtain precise image locations without going through lengthy two-dimensional convolutions. Furthermore we use a measurement noise calculated from the covariance of the line segment fitting process. The failure to find certain vertices is detected and indicates either an occlusion not covered by the occlusion reasoning step (described in the next subsection), or a motion not covered by our motion model.

4.3 Occlusion Reasoning and Re-Initialization

Tracking is performed on certain artificial landmarks in the scene. Their visibility can only be corrupted if their image projection falls outside the field of view of the camera or if they are occluded by other *real* objects. Full 3D occlusion reasoning requires monitoring all moving objects and knowledge about the entire 3D geometry, a feature not yet implemented

in our system. Figure 6 (a) illustrates an occlusion example. We are currently investigating the use of additional features, such as arbitrary corners or edges which will be added once the tracker has been initialized from the known landmarks.

Failure to find certain landmarks is indicated by a very large measurement noise. Such *unreliable* landmark points are discounted by the Kalman filter. If too many landmark points are labelled as *unreliable*, the tracker re-initializes itself by re-calibration.

5 Results

The system is currently implemented on Silicon Graphics workstations using SGI's ImageVision and VideoLibrary as well as Performer and OpenGL. It successfully tracks landmarks and estimates camera parameters at approximately 10 Hz with a live PAL-size video stream on a Silicon Graphics Indy.

Our landmarks are black cardboard squares placed on a wall, as seen in Figures 5–6. In a first set of experiments with an initial version of our tracker we recorded an image sequence from a moving camera pointing at the wall. Virtual furniture is then overlaid according to the estimated camera parameter (cf. Figure 5). Since we have a 3D representation of the room and the camera, we are able to perform collision detection between the furniture and the room [14]. The user places the virtual furniture in the augmented scene by interactively pushing it to the wall until a collision is detected. The AR system then automatically lowers the furniture until it rests on the floor.

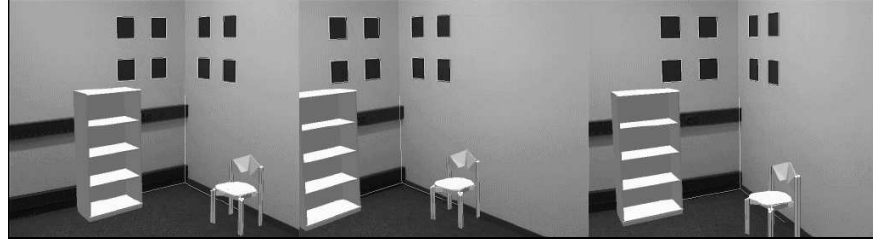


Fig. 5: Images of a sequence with overlaid virtual furniture. The estimated position of the world coordinate axes is also overlaid on the room corner.

Figure 6 shows various screen-shots from the video screen of the system running in real-time. The figures also exhibit some possible AR applications: 6 b) shows an additional virtual room divider and a reference floor grid; 6 c) visualizes the usual invisible electrical wires inside the wall; 6 d) shows the fire escape routes; 6 e) a (red) arrow (right) shows where to find the fire alarm button, and 6 f) explicitly shows the fire hose as a texture mapped photo of the inside of a cabinet.

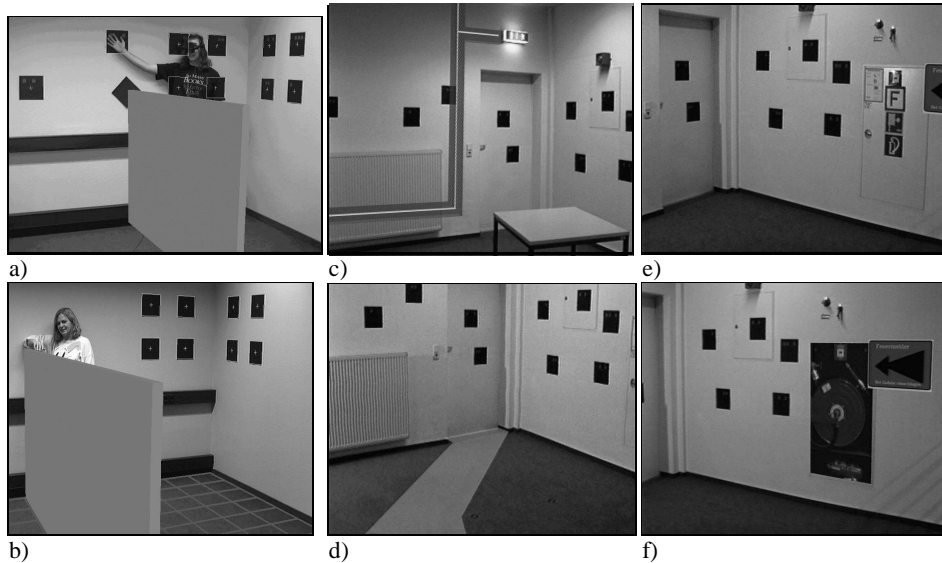


Fig. 6: (a) We successfully track with partial occlusion as long as at least two landmarks (squares) are visible. Models of the occluded landmark as well as a virtual divider have been overlaid to the video. The next images exhibit various AR applications: (b) a virtual room divider and floor grid, (c) electric wires inside the wall, (d) a fire escape route is being shown, (e) a (red) arrow (right) shows where to find the fire alarm button, (f) like (e), but a texture mapped photo of the inside of a cabinet has been superimposed on the cabinet door.

6 Conclusion

In this paper we addressed two major problems of AR applications: (a) the precise alignment of real and virtual coordinate frames for overlay, and (b) capturing the 3D motion of a camera including camera position estimates for each video frame. The latter is especially important for interactive AR applications, where users can manipulate virtual objects in an augmented real 3D environment. This problem has not been tackled successfully before using only video-input measurements.

Intrinsic and extrinsic camera parameters of a real camera are estimated using an automated camera calibration procedure based on landmark detection. These parameter sets are used to align and overlay computer generated graphics of virtual objects onto live video. Since extrinsic camera parameters are estimated separately the virtual objects can be manipulated and placed in the real 3D environment including collision detection with the room boundary or other objects in the scene. We furthermore apply extended Kalman filter techniques for estimating the motion of the camera and the extrinsic camera parameters. Due to the lack of knowledge about the camera movements produced by the user, we simply impose an acceleration-free constant angular velocity and constant linear acceleration-motion to the camera. Angular accelerations and linear jerk caused by the user moving the camera are successfully modeled as process noise.

Robustness has been achieved by using model-driven landmark detection and landmark tracking instead of pure data-driven motion estimation. Real-time performance on an entry level Silicon Graphics workstation (SGI Indy) has been achieved by carefully evaluating each processing step and using lightweight landmark models as tracking features, as well as, well designed image measurement methods in the Kalman filter. The system successfully tracks landmarks and estimates camera parameters at approximately 10 Hz with a live PAL-size video stream on a Silicon Graphics Indy.

Future work will include a fusion of model- and data-driven feature tracking in order to improve performance along occlusions and to expand the allowed camera motion.

7 Acknowledgments

We would like to thank K. Ahlers, C. Crampton, and D.S. Greer of the former UI&V group of ECRC for their help in building our AR system. One of us (D.K.) acknowledges P. Perona (Caltech) for financial support during his stay at Caltech.

This research has been financially supported in part by Bull SA, ICL Plc, Siemens AG, and by the European Community under ACTS Project # AC017 (Collaborative Integrated Communications for Construction).

References

1. M. Bajura, H. Fuchs, and R. Ohbuchi, "Merging virtual objects with the real world: Seeing ultrasound imagery within the patient," in *Computer Graphics (SIGGRAPH '92 Proceedings)* (E. E. Catmull, ed.), vol. 26(2), pp. 203–210, July 1992.
2. W. Lorensen, H. Cline, C. Nafis, R. Kikinis, D. Altobelli, and L. Gleason, "Enhancing reality in the operating room," in *Visualization '93 Conference Proceedings*, (Los Alamitos, CA), pp. 410–415, IEEE Computer Society Press, October 1993.
3. A. State, M. Livingston, W. Garrett, G. Hirota, M. Whitton, E. Pisano, and H. Fuchs, "Technologies for augmented reality systems: Realizing ultrasound-guided needle biopsies," in *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '96* (New Orleans, LA), pp. 439–446, ACM SIGGRAPH, New York, August 1996.
4. W. Grimson, T. Lozano-Perez, W. Wells, G. Ettinger, and S. White, "An automatic registration method for frameless stereotaxy, image, guided surgery and enhanced reality visualization," in *IEEE Conf. Computer Vision and Pattern Recognition*, (Seattle, WA, June 19–23), pp. 430–436, 1994.
5. T. Caudell and D. Mizell, "Augmented reality: An application of heads-up display technology to manual manufacturing processes," in *Proceedings of Hawaii International Conference on System Sciences*, pp. 659–669, January 1992.
6. P. Milgram, S. Zhai, D. Drascic, and J. Grodski, "Applications of augmented reality for human-robot communication," in *Proceedings of IROS '93: International Conference on Intelligent Robots and Systems*, (Yokohama, Japan), pp. 1467–1472, July 1993.
7. C. Chevrier, S. Belblidia, and J. Paul, "Composing computer-generated images and video films: An application for visual assessment in urban environments," in *Computer Graphics: Developments in Virtual Environments (Proceedings of CG International '95 Conference)*, (Leeds, UK), pp. 115–125, June 1995.
8. E. Rose, D. Breen, K. Ahlers, C. Crampton, M. Tuceryan, R. Whitaker, and D. Greer, "Annotating real-world objects using augmented reality," in *Computer Graphics: Developments in Virtual Environments (Proceedings of CG International '95 Conference)*, (Leeds, UK), pp. 357–370, June 1995.

9. K. Ahlers, A. Kramer, D. Breen, P. Chevalier, C. Crampton, E. Rose, M. Tuceryan, R. Whitaker, and D. Greer, "Distributed augmented reality for collaborative design applications," in *Eurographics '95 Proceedings*, (Maastricht, NL), pp. 3–14, Blackwell Publishers, August 1995.
10. G. Klinker, K. Ahlers, D. Breen, P.-Y. Chevalier, C. Crampton, D. Greer, D. Koller, A. Kraemer, E. Rose, M. Tuceryan, and R. Whitaker, "Confluence of computer vision and interactive graphics for augmented reality," *Presence: Teleoperators and Virtual Environments (Special issue on Augmented Reality)*, January 1997.
11. A. State, M. Livingston, W. Garrett, G. Hirota, M. Whitton, E. Pisano, and H. Fuchs, "Superior augmented reality registration by integrating landmark tracking and magnetic tracking," in *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '96* (New Orleans, LA), pp. 429–438, ACM SIGGRAPH, New York, August 1996.
12. M. Uenohara and T. Kanade, "Vision-based object registration for real-time image overlay," *Computers in Biology and Medicine*, vol. 25, pp. 249–260, March 1995.
13. K. Kutulakos and J. Vallino, "Affine object representations for calibration-free augmented reality," in *Virtual Reality Ann. Int'l Symposium (VRAIS '96)*, pp. 25–36, 1996.
14. D. Breen, R. Whitaker, E. Rose, and M. Tuceryan, "Interactive occlusion and automatic object placement for augmented reality," in *Eurographics '96 Proceedings*, (Poitiers, France), pp. 11–22, Elsevier Science Publishers B.V, August 1996.
15. J. Mellor, "Realtime camera calibration for enhanced reality visualization," in *First Int'l Conf. on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed)*, (Nice, France, April 3–6, 1995, N. Ayache (ed.), Lecture Notes in Computer Science **905**, Springer-Verlag, Berlin, Heidelberg, New York), 1995.
16. M. Bajara and U. Neumann, "Dynamic registration correction in video-Based augmented reality systems," *IEEE Computer Graphics and Applications*, vol. 15, pp. 52–61, September 1995.
17. D. Gennery, "Tracking known three-dimensional objects," in *Proc. Conf. American Association of Artificial Intelligence*, (Pittsburgh, PA, Aug. 18-20), pp. 13–17, 1982.
18. D. Lowe, "Robust model-based motion tracking through the integration of search and estimation," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 113–122, 1992.
19. D. Gennery, "Visual tracking of known three-dimensional objects," *International Journal of Computer Vision*, vol. 7, pp. 243–270, 1992.
20. Z. Zhang and O. Faugeras, *3D Dynamic Scene Analysis*. No. 27 in Springer Series in Information Science, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1992.
21. G. Adiv, "Determining 3-d motion and structure from optical flow generated by several moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, pp. 384–401, 1985.
22. T. Huang, "Determining three-dimensional motion and structure from perspective views," in *Handbook of Pattern Recognition and Image Processing*, pp. 333–354, 1986.
23. T. Broida, S. Chandrashekar, and R. Chellappa, "Recursive 3-d motion estimation from a monocular image sequence," *IEEE Trans. Aerospace and Electronic Systems*, vol. 26, pp. 639–656, 1990.
24. Z. Zhang, "Estimating motion and structure from correspondences of line segments between two perspective images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 1129–1139, December 1995.
25. H. Shariat and K. Price, "Motion estimation with more than two frames," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-12, pp. 417–434, 1990.
26. J. Weng, P. Cohen, and M. Herniou, "Calibration of stereo cameras using a non-linear distortion model," in *Proc. Int. Conf. on Pattern Recognition*, (Atlantic City, NJ, June 17-21), pp. 246–253, 1990.
27. M. Tuceryan, D. Greer, R. Whitaker, D. Breen, C. Crampton, E. Rose, and K. Ahlers, "Calibration requirements and procedures for a monitor-based augmented reality system," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 3, pp. 255–273, 1995.
28. J. Barrera, J. Banon, and R. Lotufo, "Mathematical morphology toolbox for the khoros system," in *Conf. on Image Algebra and Morphological Image Processing V, International Symposium on Optics, Imaging and Instrumentation, SPIE's Annual Meeting*, (24-29 July, 1994, San Diego, USA), 1994.
29. H. Goldstein, *Classical Mechanics*. Reading, MA: Addison-Wesley Press, 1980.
30. D. Koller, "A robust vision-based tracking technique for augmented reality applications." in preparation, 1997.
31. A. Gelb, ed., *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.
32. Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Boston, London: Artech House, 1993.