To appear in the Proceedings of the Symposium on Virtual Reality Software and Technology (VRST-97),
Lausanne, Switzerland, September 15–17, 1997

# Real-time Vision-Based Camera Tracking for Augmented Reality Applications

Dieter Koller[1,2,3], Gudrun Klinker[1], Eric Rose[1], David Breen[4], Ross Whitaker[5], and Mihran Tuceryan[6]

[1] Fraunhofer Project Group for Augmented Reality 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

*Augmented reality deals with the problem of dynamically augmenting or enhancing (images or live video of) the real world with computer generated data (e.g., graphics of virtual objects). This poses two major problems: (a) determining the precise alignment of real and virtual coordinate frames for overlay, and (b) capturing the 3D environment including camera and object motions. The latter is important for interactive augmented reality applications where users can interact with both real and virtual objects.*

*Here we address the problem of accurately tracking the 3D motion of a monocular camera in a known 3D environment and dynamically estimating the 3D camera location. We utilize fully automated landmark-based camera calibration to initialize the motion estimation and employ extended Kalman filter techniques to track landmarks and to estimate the camera location. 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.*

## 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 virtual geometric objects placed into the environment, or a display of non-geometric information about existing real objects. AR allows a user to work with and examine real 3D objects while visually receiving additional computer-based information about those objects or the task at hand. By exploiting people's visual and spatial skills, AR brings information into the user's *real* world rather than forcing the user into the computer's *virtual* world. Using AR technology, users may therefore 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 [Bajura *et al.* 92; Lorensen *et al.* 93; State *et al.* 96a; Grimson *et al.* 94]. For manufacturing AR is being used to direct workers wiring an airplane [Caudell & Mizell 92]. In telerobotics AR provides additional spatial information to the robot operator [Milgram *et al.* 93]. AR may also be used to enhance the lighting of an architectural scene [Chevrier *et al.* 95], as well as, provide part information to a mechanic repairing an engine [Rose *et al.* 95]. For interior design AR may be used to arrange virtual furniture in a real room [Ahlers *et al.* 95]. The application that is currently driving our research in augmented reality involves merging CAD models of buildings with video acquired at a construction site in real-time.

### 1.1 Augmented Reality Technical Problems

A number of technical problems must be addressed in order to produce a useful and convincing video-based augmented reality system:

1. A video-based AR system essentially has two cameras, a real one which generates video of the real environment, and a virtual one, which generates the 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. To achieve this, an initial calibration of the real camera and a dynamic update of its external parameters are required.

2. In order to have correct interactions between real and virtual objects in an AR environment, precise descriptions of the shape and location of the real objects in the environment must be acquired. These interactions may include collision detection, dynamic responses and visual occlusions [Breen *et al.* 96]. These effects require

an initial calibration/registration of models to objects and the subsequent dynamic update of these models based on tracking the corresponding real objects. The general shape of the environment may also be directly acquired with a variety of techniques (e.g. shape-from-shading, [Oliensis & Dupuis 93; Ikeuchi & Horn 81]).

3. Correct lighting is an essential part of generating virtual objects with convincing shading. It is therefore important to properly model the lighting of a real environment and project it onto the virtual objects. It is equally important and difficult to modify the shading of real objects within the video stream with virtual light sources [Chevrier *et al.* 95; Fournier 94].

4. An augmented reality system should interactively provide user requested information. Since the user is working in an actual 3D environment, the system should receive information requests through non-conventional means, either by tracking the motions of the user and interpreting her/his gestures, or through a speech recognition system.

5. The information displayed in and merged with the real environment must effectively communicate key ideas to the user. Therefore data visualization techniques within this new paradigm that effectively present data in a 3D setting need to be developed.

### 1.2 Technical Contribution

Our target application involves tracking a camera moving around a construction site. We focused primarily on vision-based algorithms for determining the position and orientation of the camera, addressing item #1 in the previous list, because these algorithms should give us the most flexibility when dealing with the diverse environments present on construction sites. Magnetic tracking devices being used in other augmented reality applications (like in [Rose *et al.* 95; State *et al.* 96b]) are not feasible in such a scenario, mainly because of (a) their limited range (3–5m), (b) interference with ferromagnetic objects of the environment, and (c) their lack of portability. Magnetic tracking also requires more initial calibration. However, vision-based tracking is computationally more expensive than magnetic-based tracking.

In this paper we specifically focus on the problem of accurately tracking the motion of a monocular camera in a known 3-D environment based on video-input only. Since we initially plan to place known landmarks within the construction sites, our first experiments search for and track the corners of rectangular patterns attached to a wall. Tracking of these corner points is based on extended Kalman filter techniques using an acceleration-free constant angular velocity and constant linear acceleration motion model. Angular accelerations and linear jerk are successfully modeled as process noise. We demonstrate the robustness and accuracy of our tracker within an augmented reality interior design application, which may also be used for exterior construction site applications.

### 1.3 Related Work

A number of groups have explored the topic of camera tracking for augmented reality. Vision-based object registration and tracking for real-time overlay has been demonstrated by [Uenohara & Kanade 95]. Their approach, however, is not effective for *interactive* augmented reality, since it does not address the complete 3D problem. It directly computes the image overlay instead of utilizing a pose calculation based image overlay. A pose calculation is, however, necessary for *interactive* augmented reality, where real and virtual objects interact, as in [Breen *et al.* 96], and hence camera pose and object pose need to be kept decoupled and computed separately. A similar approach has been reported by [Mellor 95] in the context of enhanced reality in medicine [Grimson *et al.* 94], where near real-time calibration is performed for each frame based on a few fiducial marks. However, as in the previous approach they solve only for the complete transformation from world to image points instead of the separate extrinsic and intrinsic parameter estimates necessary for interactive augmented reality applications.

Kutulakos et al. [Kutulakos & Vallino 96] solve a simliar problem like ours. By using an affine representation for coordinates and a transformation with a weak perspective approximation they avoid an initial calibration and pose reconstruction. Because of the weak perspective approximation, however, they experience limited accuracy, especially for environments with significant depth extent, where the weak perspective approximation is violated. (They are currently investigating a full perspective version.) They also use artificial fiducial marks for (affine) tracking. However, they require the user to interactively select at least four non-coplanar points as a bootstrap procedure, whereas our approach allows automatic feature selection and automatic initial calibration.

Some researcher [Uenohara & Kanade 95; Kutulakos & Vallino 96] 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 informations is necessary. For generic applications, however, we prefer the more complex pose calculation based approach which allows the decomposition of the image transformation into camera/object pose and the full perspective projection matrix. This then poses no constraints in applying standard interaction methods, like collision or occlusion detection.

Work closely related to our approach is also described in [State *et al.* 96b; Bajura & Neumann 95], where a hybrid vision and magnetic system is employed to improve the accuracy of tracking a camera over a wide range of motions and conditions. They show an accuracy typical for vision applications combined with the robustness of magnetic trackers. Their hybrid approach only works within the restricted area of a stationary magnetic tracker. While our approach is being developed to work with a mobile camera scanning an outdoor construction site.

Tracking known objects in 3D space and ego-motion es-

timation (camera tracking) have a long history in computer vision (e.g. [Gennery 82; Lowe 92; Gennery 92; Zhang & Faugeras 92]). 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. [Adiv 85]) as well as from discrete moving image features like corners or line segments (e.g. [Huang 86; Broida *et al.* 90; Zhang 95]), often coupled with structure-from-motion estimation, or using more than two frames (e.g. [Shariat & Price 90]). 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 work described in [Zhang & Faugeras 92].

### 1.4 Outline of the Paper

We start with the camera calibration procedure described in Section 2. In Section 3 we explain the motion model employed in our Kalman filter based tracking procedure, which is then described in Section 4. We finally present our initial results in Section 5 and close with a conclusion in Section 6.

## 2 Camera Calibration

The procedure of augmenting a video frame by adding a rendered virtual object requires an accurate alignment of coordinate frames, in which the real and virtual objects are represented, and other rendering parameters, e.g., internal camera parameters.

Internal, as well as, external camera parameters are determined by an automated (i.e. with no user interaction) camera calibration. The internal parameters, focal length and focal center ($f_x, f_y, c_x, c_y$). are based on the standard pinhole camera model with no lens distortion[1], and are fixed during a session. The external parameters describe the transformation (rotation and translation) from world to camera coordinates and undergo dynamic changes during a session (e.g., camera motion).

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 a slightly engineered environment. We attempt to find the image locations of markers placed in the 3D environment at known 3D locations (cf. Figure 4). This addresses the trade-off between high precision calibration and minimal or no user interaction. In the first step we locate these markers in the image through extracting the centers of dark blobs and use it as a rough initial calibration. 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 [Weng *et al.* 90] and implemented in [Tuceryan *et al.* 95].

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. [Barrera *et al.* 94]). 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, which represent binary encodings of the identification numbers of the model squares (cf. Fig. 2). 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 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 7 a)).

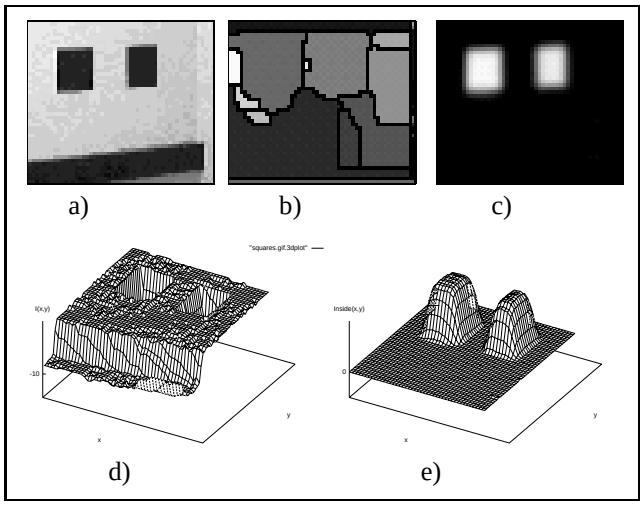


**Figure 1:** (a) Subimage with dark squares, (b) watershed transformation with greycoded regions (waterhsed are drawn in black), (c) result of the greyscale inside operation for the regions of (b), measuring the depth of *puddles* — the dark squares provide a strong filter response. (d) and (e) show 3D plots of images (a) and (c), respectively.

## 3 Motion Model For Rigid Body Motion

Any tracking approach requires some kind of motion model, even if it is constant motion. Our application scenario suggests a fairly irregular camera and object motion within all 6 degrees of freedom[2]. Since we have no *a priori*

[1] The reason for not compensating for lens distortion is that we are using the workstation's graphics pipeline for display, which does not allow for lens distortion in its rendering, besides corrections through real-time image warping using real-time texture mapping.

[2] 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.

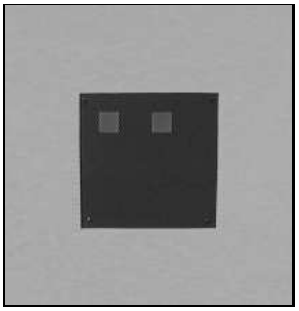

**Figure 2:** Closeup of one black calibration square exhibiting the internal smaller (red) squares used to determine the squares ID (cf. text).

knowledge about the forces changing the motion of the camera or the objects, we assume no forces (accelerations) and hence constant velocities. It is well known that in this case a general motion can be decomposed into a constant translational velocity $\boldsymbol{v}_c$ at the center of mass $\boldsymbol{c}$ of the object, and a rotation with constant angular velocity $\boldsymbol{\omega}$ around an axis through the center of mass (cf. Figure 3 and [Goldstein 80]).

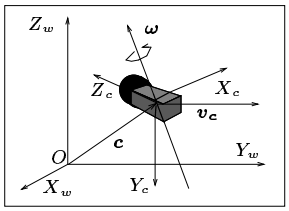


**Figure 3:** Each 3D motion can be decomposed into a translational velocity $\boldsymbol{v_c}$ and a rotation $\boldsymbol{\omega}$ about an axis through the center of mass $\boldsymbol{c}$ of the object, which is constant in the absence of any forces. $(X_w, Y_w, Z_w)$ denotes the world coordinate frame, and $(X_c, Y_c, Z_c)$ denotes the camera coordinate frame.

The motion equation of a point $\boldsymbol{p}$ on the object is then given by:

$$\dot{\boldsymbol{p}} = \boldsymbol{v}_c + \boldsymbol{\omega} \times (\boldsymbol{p} - \boldsymbol{c}), \tag{1}$$

where $\times$ denotes the cross or wedge product. Since $\boldsymbol{c}$ itself is moving, the center of rotation is also moving. If we represent the rotation with respect to the world frame origin ($\boldsymbol{c} = \boldsymbol{0}$ in Eqn. 1) then the two *motion parameters*, rotation and translation, are no longer constant for a rigid body motion with constant rotation $\boldsymbol{\omega}$ and translation $\boldsymbol{v}_c$ with respect to object coordinates. Instead if we substitute $\boldsymbol{c}(t) = \boldsymbol{c}(t_0) + \boldsymbol{v}_c(t - t_0)$ we produce the motion equation:

$$\dot{\boldsymbol{p}}(t) = \boldsymbol{v} + \boldsymbol{\omega} \times \boldsymbol{p} + \boldsymbol{a}\, t \tag{2}$$

with $\boldsymbol{v}(t_0) = \boldsymbol{v}_c - \boldsymbol{\omega} \times \boldsymbol{c}(t_0)$ and $\boldsymbol{a} = -\boldsymbol{\omega} \times \boldsymbol{v}_c = \text{const}$. The rotation is now with respect to world coordinates. However, an additional acceleration term $\boldsymbol{a}$ is added. But it has been shown in [Zhang & Faugeras 92] that as long as $\boldsymbol{\omega}$ is constant and the velocity $\boldsymbol{v}$ can be written in orders of $(t - t_0)$, Eqn. 2 is still integrable, an important fact being used in the prediction step of the Kalman filter (cf Section 4). The integration yields (cf. [Zhang & Faugeras 92; Koller 97]):

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

with

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

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

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

This motion model describes a constantly rotating and translating object in world coordinates (e.g., the position of the valve of a rotating wheel describes a cycloid curve in world coordinates). In fact, $R(\boldsymbol{\theta})$ is the Rodrigues formula of a rotation matrix according to the rotation given by the rotation vector $\boldsymbol{\theta}$. Here we use the rotation vector representation with $\boldsymbol{\theta} = \boldsymbol{\omega}\,\Delta t = (\theta_x, \theta_y, \theta_z)$, $\theta = ||\boldsymbol{\theta}||$, and $\Theta$ the skew-symmetric matrix to the vector $\boldsymbol{\theta}$:

$$\Theta = \begin{pmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{pmatrix}$$

## 4 Camera Tracking

Calibration and registration refer to the stationary aspects of a scene. In a general AR scenario, however, we must deal with wanted and unwanted dynamic scene changes. With tracking our system is able to cope with dynamic scene changes. If the external camera parameters and the objects' pose are the results of the calibration and registration procedure, respectively, then tracking can be regarded as a continuous update of those parameters.

All vision-based tracking methods are based on detecting and tracking certain features in images. These can be lines, corners, or any other salient features, which are easily and reliably detected in the images and can be uniquely associated with features of the 3D world. Our tracking approach currently uses the corners of squares attached to moving objects or walls (cf. Figure 4), which have already been used for camera calibration.

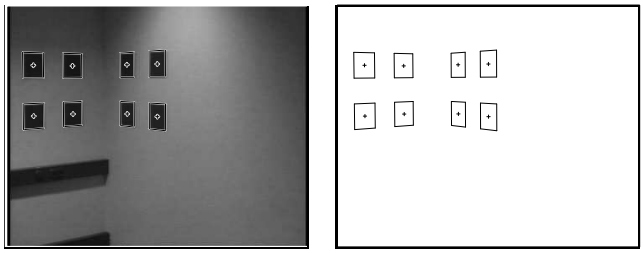

**Figure 4:** Our vision-based tracking approach currently tracks the corners of squares. The left figure shows a corner of a room with eight squares. The right figure shows the detected squares only.

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. 2.

### 4.1 Extended Kalman Filter

Our state vector $\boldsymbol{s}$ of the Kalman filter comprises the following 15 components: the position vector $\boldsymbol{t}$, the rotation vector $\boldsymbol{\phi}$, the translational and angular velocity $\boldsymbol{v}$ and $\boldsymbol{\omega}$, respectively, and the translational acceleration $\boldsymbol{a}$:

$$\boldsymbol{s} = \{\boldsymbol{t}, \boldsymbol{\phi}, \boldsymbol{v}, \boldsymbol{\omega}, \boldsymbol{a}\}.$$

We use the 3-dimensional rotation vector representation $\boldsymbol{\phi}$ for parameterizing the rotation. This way we can avoid applying additional constraints in the minimization which are required when using the redundant 4 parameter Quaternion representation.

We do not include the extended Kalman filter (EKF) equations since they can be found in most related textbooks, e.g., [Gelb 74]. 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. [Bar-Shalom & Li 93]):

$$P_k^{+\,-1} = P_k^{-\,-1} + H_k^T R_k^{-1} H_k, \tag{3}$$

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 $\hat{\boldsymbol{s}}_k^+$ then becomes:

$$\hat{\boldsymbol{s}}_k^+ = \hat{\boldsymbol{s}}_k^- + K_k(\boldsymbol{z}_k - \boldsymbol{h}_k(\hat{\boldsymbol{s}}_k^-)) \quad \text{with} \tag{4}$$

$$K_k = \left(P_k^{-\,-1} + H_k^T R_k^{-1} H_k\right)^{-1} H_k^T R_k^{-1}, \tag{5}$$

which requires the inverse of the updated covariance matrix $P_k^+$ of Eqn 3. Inverting $R_k$ is straightforward since we assume independent measurements producing a diagonal measurement noise matrix $R_k$. The transition equation (prediction) becomes $\hat{\boldsymbol{s}}_{k+1}^- = \boldsymbol{f}(\hat{\boldsymbol{s}}_k^+)$, with the transition function (using $\hat{\boldsymbol{\theta}}_k^+ = \hat{\boldsymbol{\omega}}_k^+ \Delta t$):

$$\boldsymbol{f}(\hat{\boldsymbol{s}}_k^+) = \begin{pmatrix} R(\hat{\boldsymbol{\theta}}_k^+)\,\hat{\boldsymbol{t}}_k^+ + S(\hat{\boldsymbol{\theta}}_k^+)\,\hat{\boldsymbol{v}}_k^+\,\Delta t + T(\hat{\boldsymbol{\theta}}_k^+)\,\hat{\boldsymbol{a}}_k^+\,(\frac{\Delta t}{2})^2 \\ \boldsymbol{\phi}(R(\hat{\boldsymbol{\phi}}_k^+) \cdot R(\hat{\boldsymbol{\theta}}_k^+)) \\ \hat{\boldsymbol{v}}_k^+ + \hat{\boldsymbol{a}}_k^+ \cdot \Delta t \\ \hat{\boldsymbol{\omega}}_k^+ \\ \hat{\boldsymbol{a}}_k^+ \end{pmatrix}, \tag{6}$$

according to the motion model of Eqn. 3 ($\boldsymbol{\phi}(R)$ denotes a procedures which returns the rotation vector of the rotation matrix $R$, cf. [Koller 97]).

### 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 $\boldsymbol{z}$ comprises the $x$ and $y$ image positions of all of the vertices (corners) of the $n$ squares. A measurement $\boldsymbol{z}$ is mapped to the state $\boldsymbol{s}$ by means of the measurement function $\boldsymbol{h}$: $\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{s})$.

The image corners are extracted in a multi-step procedure outlined below and in Figure 5. 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 $m_k$).

- 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 subpixel 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 image lines $l_j$ and $l_k$.

This procedure allows us to obtain precise image locations without going through lengthy two-dimensional convolutions.

Associated with the measurement is a measurement noise calculated from the covariance of the line segment fitting process. This covariance tells us how precisely an edge segment has been located and hence the precision of the associated vertex of the measurement. 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.

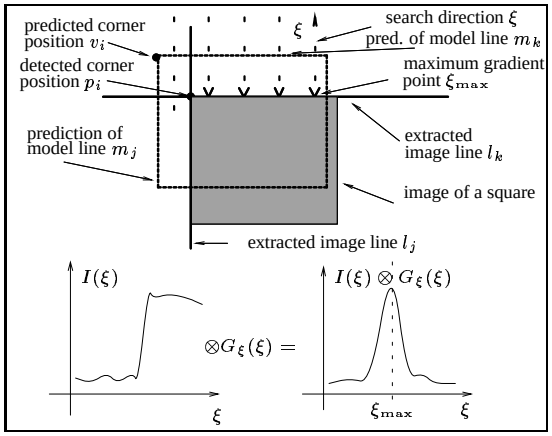


**Figure 5:** Our Kalman filter uses image corners as measurements, which are detected through intersections of matched line segments. These line segments are fitted from maximum gradient points which are produced from a one dimensional convolution with a derivative of a gaussian kernel $G_\xi$ normal to the projection of the image line ($\xi$ is a parameterization normal to the line and $I$ is the image intensity).

### 4.3 Occlusion Reasoning and Re-Initialization

Since our tracker relies only on certain artificial landmarks in the scene, it is very important to know when they

are visible. There are basically two reasons why the image measurements of the landmarks can be corrupted: (a) they are occluded by other *real* objects, (b) their image projection falls outside the field of view of the camera when the camera undergoes significant motion. Real occlusion can be detected through 3D reasoning about the scene, in which case we need to monitor all moving objects and also know the entire 3D geometry[3]. Although the second case is easily detected, it does have a major impact on the tracking algorithm. We currently only allow camera motions with at least two landmarks (squares) in the camera's field of view. Figure 7 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 itsself 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 Figure 6–7. In the first set of experiments we recorded an image sequence from a moving camera pointing at the wall. Virtual furniture is then overlayed according to the estimated camera parameter (cf. Figure 6). 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 [Breen *et al.* 96]. 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.

Figure 7 shows screen-shots from the video screen of our AR system system running in real-time. The figures also exhibit some possible AR applications: 7 a) exhibits tracking despites partial occlusions; 7 b) shows an additional virtual room divider and a reference floor grid; 7 c) visualizes the usual invisible electrical wires inside the wall; 7 d) shows the fire escape routes; 7 e) a red arrow shows where to find the fire alarm button, and 7 f) explicitly shows the fire hose as a texture mapped photo of the inside of a cabinet.

Our tracker has proven to be fairly robust with partially occluded landmarks and also with accelerated camera motions. We cannot provide quantitative tracking results since we have currently no means to record ground truth camera motions.

## 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, which is necessary for outdoor AR applications on construction sites, where magnetic tracking devices are not feasible.

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. We will also explore the possibility of fusing Global Positioning System (GPS) readings in order to assist with camera calibration and re-initialization on construction sites.

## 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.) would like to thank P. Perona (CalTech) for financial support for his stay at the California Institute of Technology.

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

[Adiv 85] G. Adiv, Determining 3-D motion and structure from optical fbw generated by several moving objects. *IEEE Transactions on*

[3] This is not yet implemented in our system.

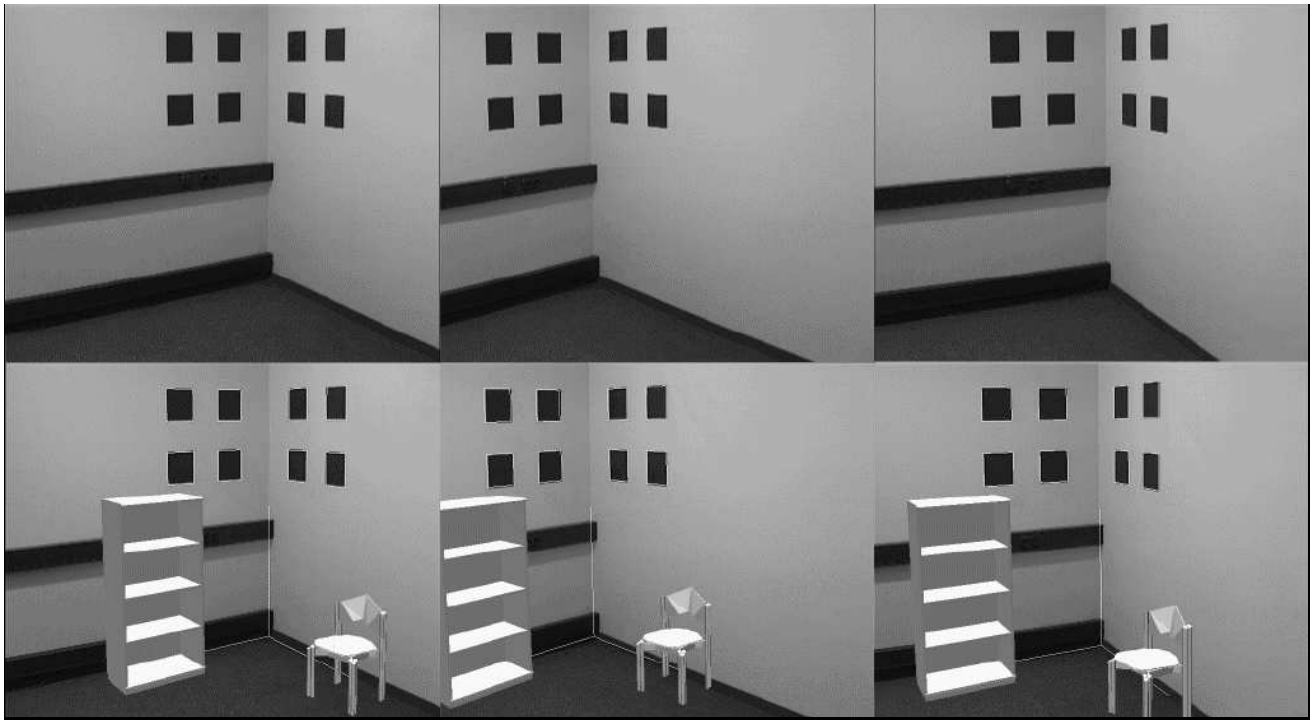

**Figure 6:** The upper row shows screen-shots from the image sequence. The lower row shows the images with overlayed virtual furniture. The estimated position of the world coordinate axes is also overlayed on the room corner.


*Pattern Analysis and Machine Intelligence* **PAMI-7**, 1985, pp. 384–401.

[Ahlers *et al.* 95] K.H. Ahlers, A. Kramer, D.E. Breen, P.-Y. Chevalier, C. Crampton, E. Rose, M. Tuceryan, R. T. Whitaker, D. Greer, Distributed Augmented Reality for Collaborative Design Applications. *Eurographics '95 Proceedings*, Blackwell Publishers, Maastricht, NL, August 1995, pp. 3–14.

[Bajura & Neumann 95] Michael Bajura, Ulrich Neumann, Dynamic Registration Correction in Video-Based Augmented Reality Systems. *IEEE Computer Graphics and Applications* **15**:5, September 1995, pp. 52–61.

[Bajura *et al.* 92] Michael Bajura, Henry Fuchs, Ryutarou Ohbuchi, Merging virtual objects with the real world: Seeing ultrasound imagery within the patient. Edwin E. Catmull (ed.): *Computer Graphics (SIGGRAPH '92 Proceedings)* **26(2)**, July 1992, pp. 203–210.

[Bar-Shalom & Li 93] Y. Bar-Shalom, X.-R. Li, *Estimation and Tracking: Principles, Technuques, and Software*. Artech House, Boston, London, 1993.

[Barrera *et al.* 94] J. Barrera, J.F. Banon, R.A. Lotufo, Mathematical Morphology Toolbox for the Khoros System. *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.

[Breen *et al.* 96] D.E. Breen, R.T. Whitaker, E. Rose, M. Tuceryan, Interactive Occlusion and Automatic Object Placement for Augmented Reality. *Eurographics '96 Proceedings*, Elsevier Science Publishers B.V, Poitiers, France, August 1996, pp. 11–22.

[Broida *et al.* 90] T.J. Broida, S. Chandrashekhar, R. Chellappa, Recursive 3-D motion estimation from a monocular image sequence. *IEEE Trans. Aerospace and Electronic Systems* **26**, 1990, pp. 639–656.

[Caudell & Mizell 92] T. Caudell, D. Mizell, Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes. *Proceedings of Hawaii International Conference on System Sciences*, January 1992, pp. 659–669.

[Chevrier *et al.* 95] C. Chevrier, S. Belblidia, J.C. Paul, Composing Computer-Generated Images and Video Films: An Application for Visual Assessment in Urban Environments. *Computer Graphics: Developments in Virtual Environments (Proceedings of CG International '95 Conference)*, Leeds, UK, June 1995, pp. 115–125.

[Fournier 94] A. Fournier, Illumination Problems in Computer Augmented Reality. *Journée INRIA, Analyse/Syntèse D'Images*, January 1994, pp. 1–21.

[Gelb 74] A. Gelb (ed.), *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.

[Gennery 82] D.B. Gennery, Tracking known three-dimensional objects. *Proc. Conf. American Association of Artificial Intelligence*, Pittsburgh, PA, Aug. 18-20, 1982, pp. 13–17.

[Gennery 92] D.B. Gennery, Visual tracking of known three-dimensional objects. *International Journal of Computer Vision* **7**, 1992, pp. 243–270.

[Goldstein 80] H. Goldstein, *Classical Mechanics*. Addison-Wesley Press, Reading, MA, 1980.

[Grimson *et al.* 94] W. Grimson, T. Lozano-Perez, W. Wells, G. Ettinger, S. White, An Automatic Registration Method for Frameless Stereotaxy, Image, Guided Surgery and Enhanced Reality Visualization. *IEEE Conf. Computer Vision and Pattern Recognition*, Seattle, WA, June 19-23, 1994, pp. 430–436.

[Huang 86] T.S. Huang, Determining Three-Dimensional Motion and Structure From Perspective Views. *Handbook of Pattern Recognition and Image Processing*, 1986, pp. 333–354.

[Ikeuchi & Horn 81] K. Ikeuchi, B.K.P. Horn, Numerical Shape from Shading and Occluding Boundaries. *Artificial Intelligence* **17**, 1981, pp. 141–184.

[Koller 97] Dieter Koller, A Robust Vision-Based Tracking Technique for Augmented Reality Applications. Technical report, California Institute of Technology, 1997, in preparation.

[Kutulakos & Vallino 96] K.J. Kutulakos, J. Vallino, Affine Object Representations for Calibration-Free Augmented Reality. *Virtual Reality Ann. Int'l Symposium (VRAIS '96)*, 1996, pp. 25–36.

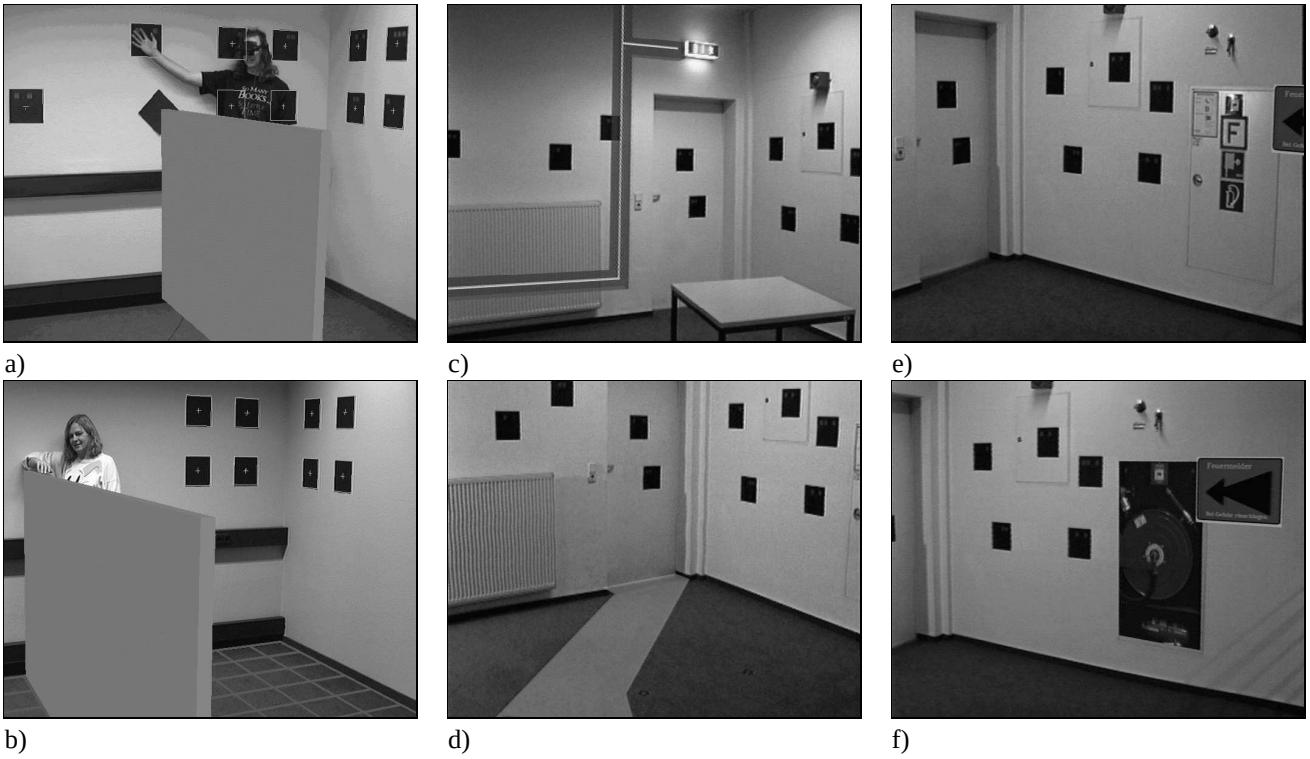


**Figure 7:** (a) We successfully track with partial occlusion as long as at least two landmarks (squares) are visible. Models of the occluded landmark as well as a virtual divider have been overlayed to the video. The next images exhibit various AR applications: (b) a virtual room divider and fbor grid, (c) electric wires inside the wall, (d) a fi re escape route is being shown, (e) a(red) arrow shows where to fi nd the fi re alarm button, (f) like (e), but a texture mapped photo of the inside of a cabinet has been superimposed on the cabinet door.


[Lorensen *et al.* 93] W. Lorensen, H. Cline, C. Nafi s, R. Kikinis, D. Altobelli, L. Gleason, Enhancing Reality in the Operating Room. *Visualization '93 Conference Proceedings*, IEEE Computer Society Press, Los Alamitos, CA, October 1993, pp. 410–415.

[Lowe 92] D. G. Lowe, Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision* **8**:2, 1992, pp. 113–122.

[Mellor 95] J.P. Mellor, Realtime Camera Calibration for Enhanced Reality Visualization. *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.

[Milgram *et al.* 93] P. Milgram, S. Zhai, D. Drascic, J.J. Grodski, Applications of Augmented Reality for Human-Robot Communication. *Proceedings of IROS '93: International Conference on Intelligent Robots and Systems*, Yokohama, Japan, July 1993, pp. 1467–1472.

[Oliensis & Dupuis 93] J. Oliensis, P. Dupuis, A Global Algorithm for Shape from Shading. *Proc. Int. Conf. on Computer Vision*, Berlin, Germany, May 11–14, 1993, pp. 692–701.

[Rose *et al.* 95] E. Rose, D. Breen, K. Ahlers, C. Crampton, M. Tuceryan, R. Whitaker, D. Greer, Annotating Real-World Objects Using Augmented Reality. *Computer Graphics: Developments in Virtual Environments (Proceedings of CG International '95 Conference)*, Leeds, UK, June 1995, pp. 357–370.

[Shariat & Price 90] H. Shariat, K.E. Price, Motion estimation with more then two frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-12**, 1990, pp. 417–434.

[State *et al.* 96a] Andrei State, Mark Livingston, William Garrett, Gentaro Hirota, Mary Whitton, Etta Pisano, Henry Fuchs, Technologies for Augmented Reality Systems: Realizing Ultrasound-Guided Needle Biopsies. *Computer Graphics Proceedings, Annual Conference Series:* SIGGRAPH '96 (New Orleans, LA), ACM SIGGRAPH, New York, August 1996, pp. 439–446.

[State *et al.* 96b] Andrei State, Mark Livingston, William Garrett, Gentaro Hirota, Mary Whitton, Etta Pisano, Henry Fuchs, Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. *Computer Graphics Proceedings, Annual Conference Series:* SIGGRAPH '96 (New Orleans, LA), ACM SIGGRAPH, New York, August 1996, pp. 429–438.

[Stuelpnagel 64] John Stuelpnagel, On the Representation of the three-dimensional Rotation Group. *SIAM Review* **6**:4, 1964, pp. 422–430.

[Tuceryan *et al.* 95] M. Tuceryan, D.S. Greer, R.T. Whitaker, D.E. Breen, C. Crampton, E. Rose, K.H. Ahlers, Calibration Requirements and Procedures for a Monitor-Based Augmented Reality System. *IEEE Transactions on Visualization and Computer Graphics* **1**:3, 1995, pp. 255–273.

[Uenohara & Kanade 95] M. Uenohara, T. Kanade, Vision-Based Object Registration for Real-Time Image Overlay. *Computers in Biology and Medicine* **25**:2, March 1995, pp. 249–260.

[Weng *et al.* 90] J. Weng, P. Cohen, M. Herniou, Calibration of stereo cameras using a non-linear distortion model. *Proc. Int. Conf. on Pattern Recognition*, Atlantic City, NJ, June 17-21, 1990, pp. 246–253.

[Zhang & Faugeras 92] Z. Zhang, O. Faugeras, *3D Dynamic Scene Analysis*. No. 27 in Springer Series in Information Science. Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1992.

[Zhang 95] Z. Zhang, Estimating Motion and Structure from Correspondences of Line Segments between Two Perspective Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**:12, December 1995, pp. 1129–1139.