Tightly Integrated Sensor Fusion for Robust Visual Tracking

Georg Klein and Tom Drummond
{gswk2|twd20}@eng.cam.ac.uk
Department of Engineering
University of Cambridge
Cambridge CB1 2PZ, UK

Abstract

This paper presents novel methods for increasing the robustness of visual tracking systems by incorporating information from inertial sensors. We show that more can be achieved than simply combining the sensor data within a statistical filter. In particular we show how, in addition to using inertial data to provide predictions for the visual sensor, this data can also be used to provide an estimate of motion blur for each feature and this can be used to dynamically tune the parameters of each feature detector in the visual sensor. This allows the system to obtain useful information from the visual sensor even in the presence of substantial motion blur. Finally, the visual sensor can be used to calibrate the parameters of the inertial sensor to eliminate drift.

1 Introduction

Visual tracking attempts to provide a real-time estimate of camera pose relative to its environment or other solid objects. This is useful for a range of applications: In manufacturing for example a camera may be mounted on a robot arm to accurately position the arm relative to a work-piece. In augmented reality, a camera may be mounted on a head-mounted-display to track the position of a user's head in the real world. In contrast to magnetic or acoustic tracking, visual tracking need not require the placement of special beacons or sensors which may limit range and be prone to interference. However, visual tracking is often not capable of tracking rapid and unpredictable camera motion such as may be produced by a hand- or head-mounted camera. This makes the addition of sensors such as inertial rate gyroscopes which are robust to large transient motions an attractive proposition.

Visual tracking systems frequently make use of fiducial markers such as LEDs, reflective markers, or distinctive colored shapes. These landmarks are placed at known positions in an environment and are easily recognized by the visual system. This approach is widely used in augmented reality systems ([7, 13, 11].) Model-based visual tracking exploits salient features already present in the object or environment to be tracked and does not require markers. Instead, a description of the salient features of the environment is used: this is most often prepared off-line and stored in a CAD model [4, 5, 8, 2] although some recent systems can generate

Figure 1: Substantial motion blur due to 2.6 rad/s camera rotation, with 100x100 pixel enlargement

models on the fly[6, 3, 10]. To achieve real-time performance, feature searches in the image are often limited to a small area around a predicted image position of a feature. As a result of this, large unpredicted image motion (e.g., if the camera is suddenly rotated) can move image features beyond range of local search. Such motions are not uncommon for a head- or hand-mounted camera. If such a camera has a focal length of 1000 pixels and is subject to a transient motion of 3 radians/s, this corresponds to 60 pixels of motion between consecutive video fields at 50Hz. Furthermore, rapid camera motion can degrade the video image due to tearing and motion-blur: this is illustrated in Figure 1 which shows the motion blur caused by a camera rotating at 2.6 radians/s. Consequently, rapid motion of the camera frequently causes visual tracking systems to fail. Inertial sensors such as rate gyroscopes and linear accelerometers provide measurements of rotational velocity and linear acceleration. Compared to video input, inertial sensors can be sampled at a high frequency and with low latency. While they are of limited suitability for directly tracking pose due to the accumulation of error in integration, their robustness to transient motion make them ideal for complementing a vision-based tracking system.

The fusion of vision and inertial measurements has been the area of substantial research, particularly in the field of augmented reality. In [13], an extended Kalman Filter combines landmark tracking and inertial navigation. A similar system is presented in [12]. A Kalman Filter represents all measurements and system state as multivariate Gaussian distributions and allows the propagation of state error estimates and optimal weighting of noisy measurements. A prerequisite for good EKF performance is the availability of good measurement and process noise models; [1] develops a system which selects one of multiple uncertainty models based on how rapidly a synthetic camera is moving. Approaching fusion from a different perspective, [9] tackles the fusion of inertial data and visual measurements of line correspondences from a control-theoretic viewpoint to produce a theoretically sound algorithm.

This paper presents a novel method of fusing rate gyroscope information with model-based tracking. We show that more can be achieved when using these kind of sensors than simply combining the data in a statistical (e.g. Kalman) filter. Figure 2 illustrates the fusion strategy employed: Rotational measurements from rate gyroscopes are used both to initialise and modify the operation of an edge-based visual sensor, while data from this sensor is used to update the inertial sensor's calibration parameters. The bold lines represent simplest possible implementation

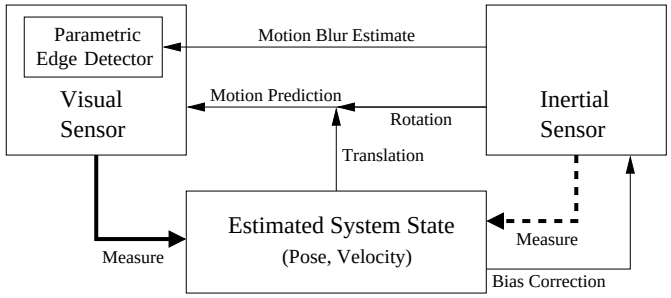

Figure 2: Sensor fusion strategy

of sensor fusion, whereby sensor measurements are only used to update the system state. In the system presented here, the dashed line is replaced by initialisation of the visual system with inertial measurements and a prior covariance matrix. Sections 2 and 3 describe the operation of the visual and inertial sensors respectively. The fusion of these two sensors is detailed in Section 4 and Sections 5 and 6 present results and conclusions.

2 Visual Sensor

2.1 Representation of Camera Pose

The visual tracking system employed is capable of tracking slow motion of a camera relative to an object. It is capable of real-time operation at 50Hz on a standard workstation. An estimate of camera pose relative to known world geometry is continually updated and described by the matrix E which transforms points in world coordinates to camera coordinates:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = E \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \tag{1}$$

E takes the form

$$E = \begin{bmatrix} & R & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where R is a rotation matrix ($|R| = 1$, $R^T R = I$) and $\mathbf{t}$ is a translation vector. The matrix E is periodically updated by means of left-multiplication by a motion matrix:

$$E_{t+1} = M_t E_t \tag{3}$$

where M takes a form similar to E. The set of all such matrices form a representation of the 6-dimensional Lie Group SE(3), the group of rigid body transformations in $\mathbb{R}^3$. The transformations possible under this group can be parametrised with a six-vector via the exponential map

$$M = \exp\left(\sum_{i=1}^{6} \mu_i G_i\right) \tag{4}$$

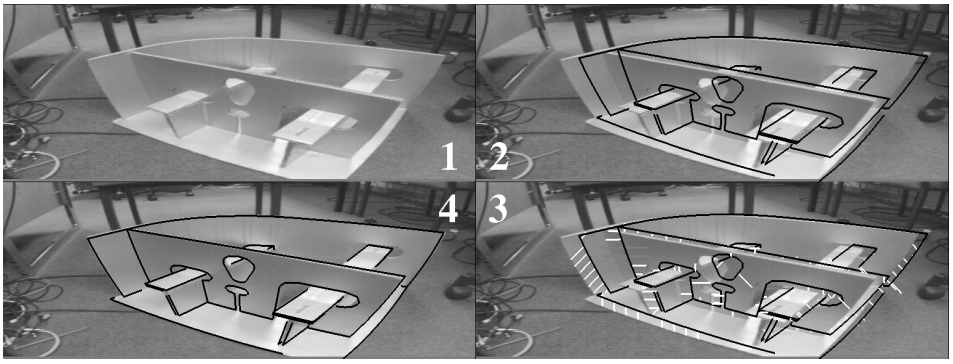

Figure 3: Tracking system loop

where G_i are the usual group generator matrices, μ_1, μ_2 and μ_3 represent translation along the x, y and z axes and μ_4, μ_5 and μ_6 describe rotation around these axes.

2.2 Visual Tracking System Loop

The visual tracking system updates the matrix E once every PAL video field (50Hz.) The steps taken each field are illustrated in figure 3, and are as follows:

Step 1: A video image is acquired from the video capture hardware,

Step 2: The model is rendered using a prediction of the camera pose,

Step 3: A local search for image edges is performed around the rendered edges,

Step 4: A motion matrix M describing the error between prediction and observation is calculated and used to update E.

Video images are captured field-by-field to avoid interlacing artefacts (tearing) and increase the tracking system's temporal resolution. Captured images are greyscale and have a resolution of 768x288. A standard CCD camera with square pixels is used. This is fitted with a wide-angle (4.2mm, f=530 pixels) lens so as to provide a large number of simultaneously visible trackable features.

The edges of the 3D model are rendered according to the current camera pose estimate. This estimate can be the previous observed pose, or it may be generated from a velocity estimate or from inertial sensors as in Section 4.1. Hidden edge removal is accomplished by OpenGL stencil buffering and BSP traversal. Since the wide-angle lens used exhibits substantial radial distortion, a standard pin-hole camera model cannot be used directly. Radial distortion is approximated by a mapping of radii in normalised camera coordinates:

$$r' = r + \alpha r^3 + \beta r^5 \tag{5}$$

where $r = \sqrt{\left(\frac{x_c}{z_c}\right)^2 + \left(\frac{x_c}{z_c}\right)^2}$. Pixel coordinates $\begin{pmatrix} u & v \end{pmatrix}^T$ are then given by

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \end{bmatrix} \begin{pmatrix} \frac{r'}{r}\frac{x_c}{z_c} \\ \frac{r'}{r}\frac{y_c}{z_c} \\ 1 \end{pmatrix} \tag{6}$$

Where f_u, f_v, u_0, v_0, α and β are camera parameters which can be calibrated on-line, with $\alpha = -0.29$ and $\beta = 0.06$ for the lens used.

Once visible model edges have been rendered, a local search for edges in the video feed is performed. For this purpose, sample points are initialised at regular intervals along model edges. One-dimensional edge detection is performed at each of these points: at the ξth sample point, edge detection in the direction of the edge normal $\hat{\boldsymbol{n}}^{\xi}$ is performed and the distance d^{ξ} to the nearest detected edge (should one be found) is determined. The edge detection process is further described in Section 4.2. Typically, several hundred sample points are initialised per video frame.

Once all edge distances have been found, the motion vector $\boldsymbol{\mu}$ which minimises the pose error is found. Equations 1 and 6 can be differentiated to find the motion in the image of each sample point with respect to the six parameters of camera motion to obtain, for the ξth sample point, the partial differentials $\frac{\partial d^{\xi}}{\partial \mu_i}$ $(1 \leq i \leq 6)$. The motion vector $\boldsymbol{\mu}$ which minimises the residual error is found by a robust re-weighted least squares method. The tracking system then updates its pose estimate E with a motion matrix corresponding to the optimal motion vector, and the tracking loop recommences.

3 Inertial Sensors

Three rate gyroscopes were affixed to the camera as shown in Figure 4. These gyroscopes produce an output voltage which varies linearly with rotational velocity. The voltage is sampled using a 10-bit ADC and transmitted to the workstation via a serial link. The sampling frequency used is 171 Hz.

The nth gyro produces an output voltage V_n:

$$V_n = B_n + \alpha_n \Omega_n \tag{7}$$

where Ω_n is rotational velocity about the nth gyroscope's axis, B_n a bias voltage and α_n the gyroscope sensitivity. At rest when $\Omega_n = 0$, B_n can be measured directly. For a single axis, angular displacement, Φ_n is found by integration:

$$\Phi_n = \int \frac{V_n - B_n}{\alpha_n} dt \tag{8}$$

The parameter α_n can be determined by performing the above integration while rotating the gyroscope about a known angle. The form of equation 8 means that

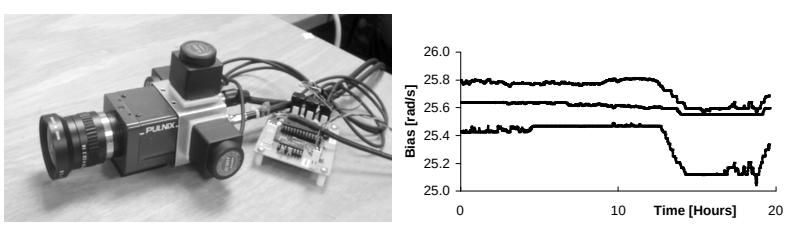

Figure 4: Rate gyroscopes affixed to camera and long-term bias drift, which can be significant with temperature change

estimates of angular position are very sensitive to errors in bias: small steady-state errors in B_n produce unbounded errors in Φ_n. Since the bias parameter for the gyros used was found to vary with time (as shown in Figure 4), the value of B_n must be continually updated to ensure long-term robustness. The mechanism used for this is described in Section 4.3.

4 Sensor Fusion

The fusion of visual and inertial sensors in our system has three key components: An initialisation of the tracking system's pose estimate before video frame processing, a modification of the edge detection process to account for motion blur, and an update of the gyroscope's bias estimate.

4.1 Visual Tracking System Initialisation

The visual tracking system described in Section 2 uses a local edge search around a predicted model position in the video feed. Furthermore, it linearises pose changes around the current pose. As a result, the visual tracking system is best suited for correcting small pose errors. If image motion beyond the range of the local edge search occurs (for example, due to sudden rapid camera rotation), the visual tracking system fails entirely.

The time needed to capture a video frame from the camera and transfer this frame from video hardware to tracking system is large compared to the time needed to sample information from the gyroscopes. Hence, a record of gyroscope information corresponding to camera motion between the previous and the current video frames is always available to the tracking system before image processing commences. This information can be used to predict camera orientation for the new video frame.

Linear accelerometers are not currently used. Instead, an estimate of linear velocity filtered from previous position measurements is stored in the system's state estimate. This velocity estimate is combined with gyroscope measurements to form the vector $\boldsymbol{\mu_p}$ representing predicted change in camera pose:

$$\boldsymbol{\mu_p} = \begin{pmatrix} x_1 & x_2 & x_3 & \Phi_1 & \Phi_2 & \Phi_3 \end{pmatrix}^T \tag{9}$$

with Φ_n evaluated as in 8 and x_n being the predicted linear displacement along the nth axis. A prediction $\hat{E}_t$ for the camera pose at time t is computed by

$$\hat{E}_t = \exp\left(-\sum_{i=1}^{6} \mu_{pi} G_i\right) E_{t-1} \tag{10}$$

This operation provides the estimate needed for step 2 in Section 2.2.

4.2 Parametric Edge detector

If the camera could be assumed to capture images using an ideal sampling function $f(t) = \delta(t)$ then edge detection could be performed by constructing a vector $\boldsymbol{i}$ of

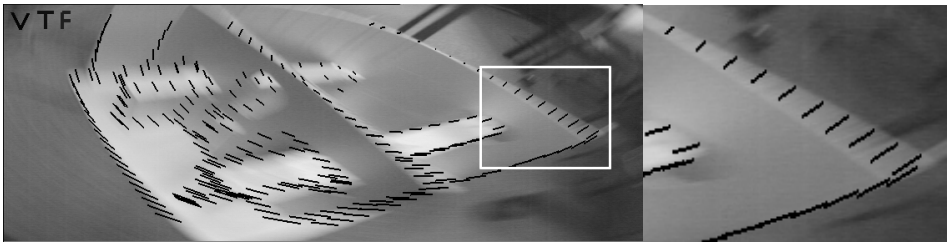

Figure 5: Image motion vectors of motion blur predicted from inertial sensors

pixel intensities around a sampling point in the direction of the edge normal and convolving with the kernel $\boldsymbol{k}_i = \begin{pmatrix} -1 & 1 \end{pmatrix}$ to give a vector of edge intensities. The values of local maxima in these intensities could be compared to a threshold value, and a suitably strong local maximum closest to the sample point is selected as the detected edge.

The assumption of a very short exposure time is however not valid for the camera used. Although cameras with very rapid exposure times exist, these usually require a high light intensity for operation and may not be suitable for operation in a standard environment, e.g. inside a normal building. Under these conditions, cameras often exhibit substantial motion blur, as illustrated in Figure 1. A better approximation of the sampling function of these cameras is a rectangular pulse:

$$f(t) = \begin{cases} \frac{1}{T_e} & -\frac{T_e}{2} \leq t \leq \frac{T_e}{2} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where T_e is the camera's exposure time. An image edge (step function) moving across the image at a rate of v pixels/second will thus appear as an intensity ramp of length vT_e pixels in the sampled field. The edge detection in step 3 of the tracking system loop can be modified to detect blurred edges by using inertial sensor information to produce an estimate of camera motion during the sampling period:

$$\boldsymbol{\mu_C} = T_e \begin{pmatrix} 0 & 0 & 0 & \Omega_1 & \Omega_2 & \Omega_3 \end{pmatrix}^T \tag{12}$$

For each sample point, an estimate of the length of an edge's motion blur in the direction of the edge normal can be found by evaluating

$$b = \left| \sum_{i=1}^{6} \mu_{Ci} \frac{\partial d}{\partial \mu_i} \right| \tag{13}$$

Figure 5 shows system's estimate of motion blur in an image superimposed over a blurred video frame. Edge detection is performed by convolution with a matched filter. The ramp kernel $\boldsymbol{k}_r$ is used, where

$$\boldsymbol{k}_r = \frac{1}{2b^2} \begin{pmatrix} -b & -b+2 & \ldots & b-2 & b \end{pmatrix} \tag{14}$$

When this kernel is convolved with the pixel intensities, the maxima indicate the sensed locations of blurred edges. The edge detection process is illustrated in Figure 6. The first plot shows image pixel intensity measured along the horizontal black line in the enlargement of Figure 7. These pixel intensities are convolved both with the ideal kernel $\boldsymbol{k_i}$ (second plot) and a ramp kernel $\boldsymbol{k}_r$ of length 36 (third plot.)

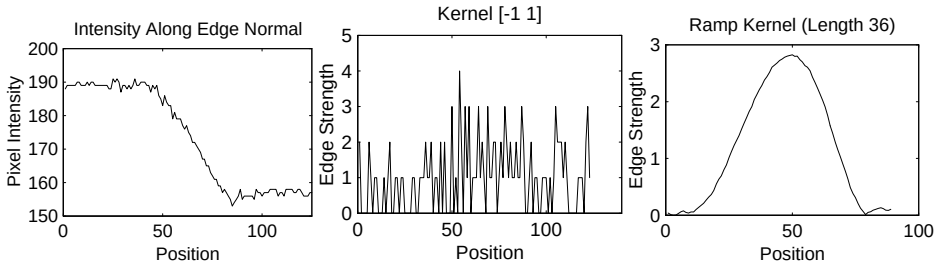

Figure 6: Plots of edge normal pixel intensity and detected edge strengths

4.3 Gyroscope Re-calibration

As shown in Figure 4, the bias parameters B_n are not constant. For long-term robustness, it is necessary to update the system's bias estimate. This is done by comparing inertial predictions of rotational motion with measurements made by the visual system. If the rotational displacement around the nth axis between two subsequent visual measurements time ΔT apart is Θ_n and the bias value used is assumed to take the form $B_n = B_n^{true} + e_n$ where e_n is bias error, it follows from equation 8 that (assuming perfect measurements)

$$e_n = \frac{\alpha_n \left(\Theta_n - \Omega_n\right)}{\Delta T} \tag{15}$$

In practice, measurements are noisy, and bias values are corrected by a small (typically $\frac{1}{100}$th) fraction of the calculated error.

5 Results

The tracking system presented was tested on three test scenes. The 'world' scene places the camera in a simple immersive table-top environment. The 'ship' scene points the camera at a model of a ship part such as could be found in a visual servoing application. The 'cubicle' scene contains a portion of a computer lab. In each scene, the camera undergoes increasingly rapid motion relative to its target while the tracking system was run in three modes: without any inertial information, using inertial information to predict camera pose, and using inertial information both for pose and blur prediction. Table 1 shows the maximum rotational velocities at which tracking was sustainable. The tracking system's performance differs greatly from scene to scene: While the 'ship' and 'world' scenes contain many edges of modest contrast, the 'cubicle' scene contains high-contrast feature such as windows and light fittings and is trackable even at high rotational velocities.

Sequence:	World	Ship	Cubicle
Visual Sensor Only [rad/s] (pixels)	0.3 (3)	0.3 (3)	1.0 (11)
Pose Initialisation Only	0.8 (8)	1.2 (13)	3.6 (38)
Motion Blur Correction	3.1 (33)	2.0 (21)	4.7 (50)

Table 1: Tracking system performance for three scenes: Maximum trackable rotational velocities in rad/s (and corresponding motion blur in pixels)

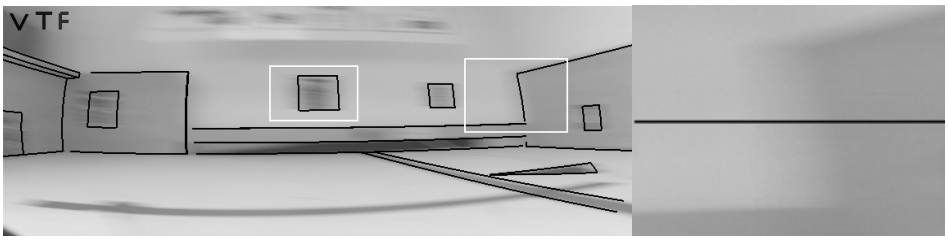

Figure 7: World sequence (with enlargement) successfully tracked at 3.1rad/s

Figure 8: Motion blur at various rotational velocities: 0.0, 0.8, 1.8, 3.1, 4.9 rad/s

Figure 7 shows the tracking system correctly tracking a sequence in the 'world' scene while the camera is rotating about its vertical axis with 3.1rad/s. This was the highest rotational velocity at which correct tracking was maintained. Video image quality at different rotational speeds is compared in Figure 8, which shows enlargements corresponding to the central white outline in Figure 7. The first four enlargements show trackable rotational velocities, the last was untrackable.

Fitting error was measured for the 'world' test scene. The mean redisual error for sample points with no motion blur was found to be 1.1 pixels. The error increased to 4.2 pixels for sample points with a motion blur of 7 pixels and reached a maximum of approximately 5.5 pixels for motion blurs of 20-33 pixels.

6 Conclusions and Further Work

This paper has presented a tightly integrated strategy for the fusion of visual and inertial sensors. The addition of an inertial pose prediction to the tracking system greatly increases the system's robustness. Pose prediction by itself is however not sufficient when camera motion is such that motion blur corrupts image measurements. In this case, the estimation of motion blur and use of a parametric edge detection algorithm further increase the robustness of the system.

While the inertial sensors used can measure rotational velocity, linear velocity is still estimated from visual measurements. Linear accelerometers will be added to the system to measure linear acceleration directly. This will increase the tracking system's robustness when the camera is undergoing translation. However, neither the rate gyroscopes nor the linear accelerometers provide any information about possible motion of the objects tracked, and so the tracking of rapidly moving objects is not supported.

Finally, the motion blur correction used is not suitable for parallel edges whose separation is comparable to the size of local motion blur. The use of more advanced rendering techniques than employed here (such as the use of multiple levels of detail) may help address this issue. This would however require suitably marked-up models, further increasing the system's already considerable dependency on data prepared off-line.

References

[1] L. Chai, K. Nguyen, W. Hoff, and T. Vincent. An adaptive estimator for registration in augmented reality. In *Proc. 2nd IEEE/ACM Int'l Workshop on Augmented Reality*, pages 20–21, October 1999.

[2] T. Drummond and R. Cipolla. Real-time tracking of complex structures with on-line camera calibration. In *Proc. British Machine Vision Conference 1999*, volume 2, pages 574–583, Nottingham, 13–16 September 1999. BMVA.

[3] V. Ferrari, T. Tuytelaars, and L. Van Gool. Markerless augmented reality with a real-time affine region tracker. In *Proc. IEEE and ACM Intl. Symposium on Augmented Reality*, volume I, pages 87–96, October 2001.

[4] C. Harris. Tracking with rigid models. In A. Blake, editor, *Active Vision*, chapter 4, pages 59–73. MIT Press, 1992.

[5] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *Int'l Journal of Computer Vision*, 29(1):5–28, 1998.

[6] B. Jiang and U. Neumann. Extendible tracking by line auto-calibration. In *Proc. IEEE and ACM Intl. Symposium on Augmented Reality*, volume I, pages 97–103. IEEE Computer Society, October 2001.

[7] A. Furhmann M. Ribo, A. Pinz. A new optical tracking system for virtual and augmented reality applications. In *Proc. IEEE Instrumentation and Measurement Technology*, volume 3, pages 1932–1936, 2001.

[8] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real-time visual tracking using a 2d-3d model-based approach. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 262–268, 1999.

[9] H. Rehbinder and B.K. Ghosh. Multi-rate fusion of visual and inertial data. In *Proc. IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 97–102, 2001.

[10] G. Simon, A. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proc. International Symposium on Augmented Reality*, pages 120–128, 2000.

[11] G. Welch and G. Bishop. SCAAT: Incremental tracking with incomplete information. *Computer Graphics*, 31(Annual Conference Series):333–344, 1997.

[12] Y. Yokokohji, Y. Sugawara, and T. Yoshikawa. Accurate image overlay on see-through head-mounted displays using vision and accelerometers. In *Proc. IEEE Conference on Virtual Reality*, pages 247–254, 2000.

[13] S. You and U. Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. In *Proc. IEEE Conference on Virtual Reality*, pages 71–78, March 2001.