# Augmented Reality Tracking in Natural Environments

Ulrich Neumann

Suya You, Youngkwan Cho, Jongweon Lee, Jun Park

Computer Science Department

Integrated Media Systems Center

University of Southern California

## Abstract

Tracking, or camera pose determination, is the main technical challenge in creating augmented realities. Constraining the degree to which the environment may be altered to support tracking heightens the challenge. This paper describes several years of work at the USC Computer Graphics and Immersive Technologies (CGIT) laboratory to develop self-contained, minimally intrusive tracking systems for use in both indoor and outdoor settings. These hybrid-technology tracking systems combine vision and inertial sensing with research in fiducial design, feature detection, motion estimation, recursive filters, and pragmatic engineering to satisfy realistic application requirements.

## 1. Introduction

Tracking, or camera pose determination, is the main technical challenge in creating augmented realities. Limiting the degree to which the environment may be modified to support tracking heightens this challenge. The USC Computer Graphics and Immersive Technologies (CGIT) laboratory strives to develop self-contained, minimally intrusive tracking systems for use in both indoor and outdoor settings. Intrusion, in the tracking context, is any type of environment modification, calibration, or constraint upon the environment *(e.g.*, placing calibrated targets or active beacons in the setting).

The goal of a self contained system leads to vision and inertial sensing subsystems that include video cameras, gyros, and accelerometers [Azum94]. Other technologies of interest for future work include RF (GPS, Impulse Radio), compass, pedometer, and altimeter sensors. No single technology is sufficiently robust and accurate (at this time) to satisfy the diversity of needs, so hybrid approaches are a rational goal [Azum95] [Baju95]. An ideal tracking system reports perfect instantaneous six-degree-of-freedom (6DOF) measurements of the sensor pose (*i.e.*, position and orientation), in any environment, under any motion. Perfect tracking is never achieved since temporal delays and accuracy limits are inherent in all measurements. The challenge in tracking research and development is to approach the ideal [Suth68] [Meye92] [Azum95] [Ward92] [Ghaz95].

## 2. Indoor Tracking

Indoor and outdoor environments present different tradeoffs. Building structures block signals used by many sensors (*e.g.*, GPS or compass). At the same time, structures provide the power and controlled environment that facilitates the installation and use of

calibrated targets or beacons. For indoor applications, a minimal set of calibrated passive targets (fiducials) appear practical and prudent if they, in turn, facilitate robust and accurate tracking. Many systems adopt this approach [Kim97] [Neum96] [Stat96] [Madr96] [Mell95] [Kutu96] [Reki97] [Klin97] [Koll97].

From a practical and engineering standpoint, colored-circle stickers make good fiducials for several reasons. They are inexpensive to produce on color inkjet or laser printers. A 2D ellipse models their projection from any viewpoint. The circles represent points, so small stickers suffice, facilitating arbitrary placement, even on small objects. Precise edge measurements are not necessary to compute an accurate centroid, reducing the influence of sensor noise and lighting variations. Color can facilitate the detection of fiducials in a scene image as well as their unique identification among the set of all fiducials.

### 2.1. Fiducial Detection

A practical indoor tracker system needs a robust method to detect fiducials [Hage96] [Heal97] [Trem97] [Kend76] [Star97] [Oakl91]. We developed two multiscale detection methods. The first approach uses calibrated color region detection and segmentation [Cho97]. The detection is straightforward, however, many variables affect the calibration including fiducial printing; camera and digitizer color response; and lighting. This approach imposes severe calibration requirements upon the user and constrains the deployment of the system. The second detection approach, based on fuzzy membership functions, addresses these deficiencies.

The fuzzy detection approach uses multiscale relationships between neighboring pixel groups to segment fiducials and their backgrounds [Lee98]. There are no threshold values to calibrate and the relationships remain stable under variations in lighting and color. After segmentation, shape and color tests distinguish fiducials from background clutter. Figure 1 shows six examples of the colored sticker fiducials.

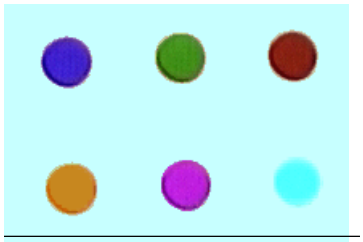


Fig. 1 – Fiducial examples based on primary and secondary colors

A fiducial is modeled as two transitions along a line, from background to fiducial color and then to background again. The transitions occur within an expectation interval determined by the range of camera-to-fiducial distances, the fiducial size, and the camera parameters. A minimum spacing between fiducials is assumed, as well as a uniform background around the fiducial. Figure 2 illustrates the fiducial detection model. The best edge position is the pixel whose intensity is the average value of its left and right segments. Pixels are grouped into R, C, and L regions according to the following restrictions and membership functions.

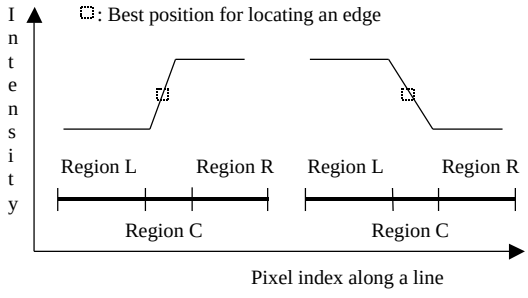



Fig. 2 – Fiducial transition model

Rule 1: Avg(R) > Avg(C) > Avg(L), or Avg(L) > Avg(C) > Avg(R), where Avg(J) is the average intensity value of region J.

Rule 2: Max(C) – Min(C) > Max(R) – Min(R) & Max(C) – Min(C) > Max(L) – Min(L), where Max(J) and Min(J) are the maximum and minimum intensity values of region J.

Rule 3: The intensity distribution of regions R and L may not overlap.

Membership functions are defined by the following equations.

$$\mu = \mu 1 \leftrightarrow \mu 2$$

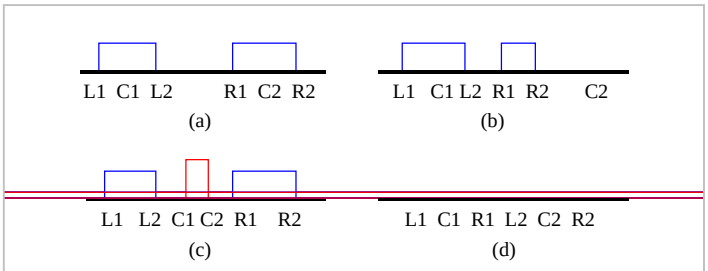
$$\mu 1 = \frac{2 \leftrightarrow MIN\left(|Avg(R) - Avg(T)|, |Avg(L) - Avg(T)|\right)}{|Avg(R) - Avg(L)|}$$

$$\mu 2 = \frac{Min(R) - Max(L)}{Max(R) - Min(L)}, \; Avg(R) > Avg(L)$$

*or*

$$\mu 2 = \frac{Min(L) - Max(R)}{Max(L) - Min(R)}, \; Avg(L) > Avg(R)$$

$\mu_1$ indicates the similarity of region T to the median intensity value of its two neighboring regions. $\mu_2$ measures the distribution similarity on both sides of a transition. An ideal noiseless transition exhibits (Min(R) – Max(L)) = (Max(R) – Min(L)) or (Min(L) – Max(R)) = (Max(L) – Min(R)), and $\mu_2 = 1$. $\mu_2 > 1$ for non ideal or noisy transitions. The peak of membership function $\mu$ locates a point closest to an ideal edge.




Note. C is a transition region
L & R are left and right homogeneous regions
1 & 2 indicate minimum and maximum intensity values of a region

Fig. 3 - Examples of relationships among transitions and left, & right regions
(a) valid fiducial
(b-d) invalid cases violating first, second, and third rules, respectively.

Figure 3 shows examples of region groupings that may arise. Only the case shown in Fig. 3a is a valid fiducial detection.

Grouping contiguous pixels that pass all restrictions identifies transitions. The transition pixel with the highest membership value is at the transition position. Connected transitions create line segments. Fiducials have solid color, so line segments with the same color should be grouped. Unfortunately, image noise invokes the need for a similarity measure. The minimum, average, and maximum pixel-intensity values of a region define a region distribution. To avoid the need for thresholds, two line segments are considered as the "same color" when their minimum and maximum intensity distributions overlap. This simple grouping function is efficient and performs well in our experiments.

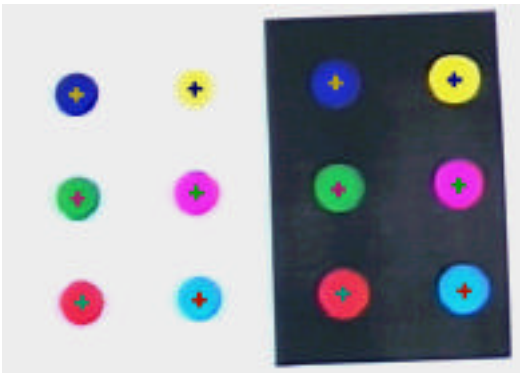
Regions that pass the color and shape tests described in [Lee98] are taken as valid fiducials. The following summarizes some test results for twelve fiducials on white and black backgrounds. Test images are captured with a Sony DXC-151A color video

camera digitized at 640×480 resolution. Two different lighting sources, daylight and fluorescent light, and a range of aperture settings (f = 1.8 ~ 8.0) of the camera create lighting variations. The algorithm detects all fiducials under every aperture settings, except for the yellow fiducial on a white background at f = 1.8 and a green fiducial on a black background at f = 8.0 and 5.6 (Table 1). The human eye does not easily perceive these undetected fiducials either.

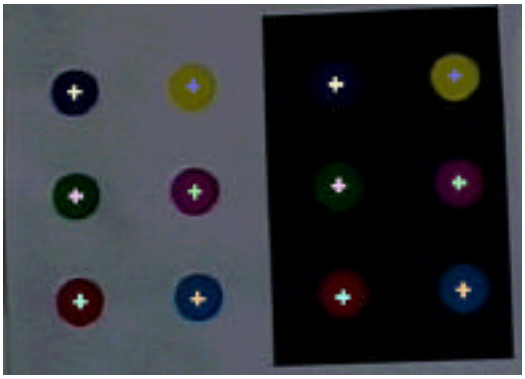
| Number of detected fiducials | | 11 | 12 | 12 | 12 | 12 | 11 | 11 | 12 | 12 | 12 | 11 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| White background | Red | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| | Green | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| | Blue | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| | Yellow | N | Y | Y | Y | Y | Y | N | Y | Y | Y | Y | Y |
| | Cyan | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| | Magenta | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| Black background | Red | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| | Green | Y | Y | Y | Y | Y | N | Y | Y | Y | Y | N | N |
| | Blue | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| | Yellow | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| | Cyan | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| | Magenta | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| Aperture (f) | | 1.8 | 2.0 | 2.8 | 4.0 | 5.6 | 8.0 | 1.8 | 2.0 | 2.8 | 4.0 | 5.6 | 8.0 |
| Lighting source | | Day light | | | | | | Fluorescent light | | | | | |

Table 1- Results of fiducial detection algorithm. Figure 4 shows example images of the experiment. Y indicates detection and N indicates no detection.

Figures 4 (a) and (b) show two images from the Table 1 tests. Figure 4 (c) and (d) show real time video images of a test scene with a more complex background and shadow patterns. In all cases the white crosses indicate a detected fiducial that also passes the shape and color tests.

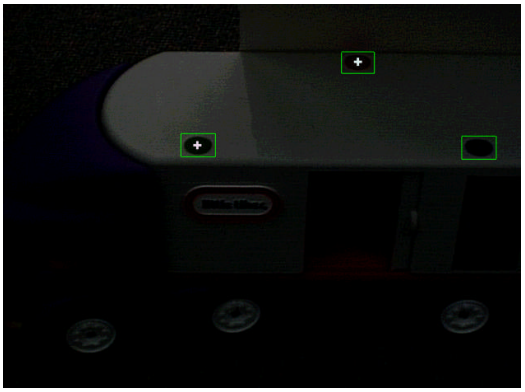


(a) Fiducials in bright light (f = 2.0)



(b) Fiducials in low light (f = 5.6)

(c) Truck scene with bright uneven lighting (f = 4.0)



(d) Truck scene with low uneven lighting (f = 8.0)

Fig. 4 - Fiducial detection under varying light conditions and backgrounds. The green rectangle without a white cross on right side of (d) indicates a detected fiducial that failed color test.

### 2.2. Scalable Fiducials

Multi-ring color fiducial systems allow for scalable numbers of fiducials and scalable tracking areas. A set of design rules describes fiducial parameter relationships that satisfy a given AR application requirement [Cho98]. In addition, the fiducial detection method converts a complex 5-DOF shape test to a series of simple problems.

When a camera is very far (or close) to a fiducial, the projected fiducial size is too small (or large) to detect it correctly. Therefore, AR systems with single-size fiducials have limited tracking range. The major axis length $d$ of a projected fiducial ellipse is $d = Df / w$, where $D$ is the diameter of the fiducial, $f$ ($= f_u = f_v$) is the effective focal length, and $w$ is the depth of the fiducial in the camera coordinate system. The tracking range could be extended by combining the detection ranges of different size ($D$) concentric rings.

Multi-ring color fiducials have varying numbers of rings, each of different sizes (Fig. 5). The first-level fiducial has one core circle and one outer ring. With increasing levels, additional rings surround the previous level fiducial. The number of rings specifies the fiducial level. Colored rings (*e.g.*, red, green, blue, yellow, magenta, and cyan) create unique patterns that facilitate fiducial identification. Figure 4 shows three size levels of

multi-ring color fiducials with two regular rules for the ring width: constant width and proportional width.

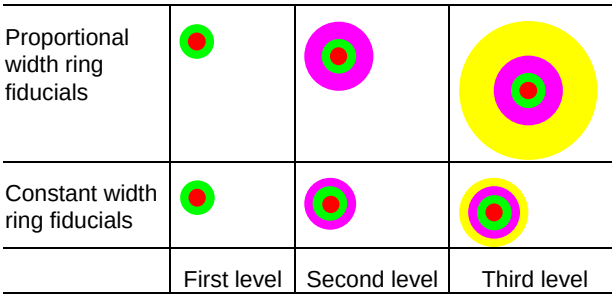



Fig. 5 – Concentric ring fiducials allow multiple levels and unique size relationships for each ring.

The proportional width system leads to scalable tracking range. The size ratio $c$ between adjacent levels is constant

$$D_i = cD_{i-1} \quad (c > 1)$$
$$= c^{i-1}D_1$$

and $D_i$ is the diameter of the level $i$ fiducial. The outer rings are more detectable from a distance, so higher level fiducials have a greater detectable range. By combining the detectable ranges of many level fiducials, we extend the system tracking range.

Let the desired tracking range be $Z_{near} \sim Z_{far}$, and the camera focal length be $f$. Let $w$ be the minimum detectable ring width in an input image. $w$ depends on the camera, the digitizer, and fiducial detection algorithm. Let the tracking range of a level $i$ fiducial be $Z_{near,i} \sim Z_{far,i}$ with the conditions $Z_{near} = Z_{near,i}$ and $Z_{far,n} = Z_{far}$. The largest detectable fiducial size in an image is $d_{near}$ ($\geq D_i f/Z_{near,i}$), and the smallest detectable fiducial size is $d_{far}$ ($\leq D_i f/Z_{far,i}$). To combine the detectable ranges smoothly, there should be no gaps between adjacent work ranges.

$$0 \leq Z_{far,i} - Z_{near,i+1} \leq \frac{D_i f}{d_{far}} - \frac{D_{i+1} f}{d_{near}} = \frac{D_i f}{d_{far}} - \frac{cD_i f}{d_{near}} = \frac{D_i f}{d_{near}}\left(\frac{d_{near}}{d_{far}} - c\right)$$

$$c \leq \frac{d_{near}}{d_{far}} \quad \text{and} \quad c \leq \frac{Z_{far,i}}{Z_{near,i}}$$

The required levels of fiducials can be expressed as a function of the size ratio $c$ for a given camera working range (Fig. 6).

$$D_n = \frac{Z_{far}}{f} d_{far}, D_1 = \frac{Z_{near}}{f} d_{near}$$

$$D_n = c^{n-1} D_1$$

$$\frac{Z_{far}}{f} d_{far} = c^{n-1} \frac{Z_{near}}{f} d_{near}$$

$$\frac{Z_{far}}{Z_{near}} = c^{n-1} \frac{d_{near}}{d_{far}} \geq c^n$$

$$n(c) \leq \frac{\log(Z_{far}/Z_{near})}{\log c}$$

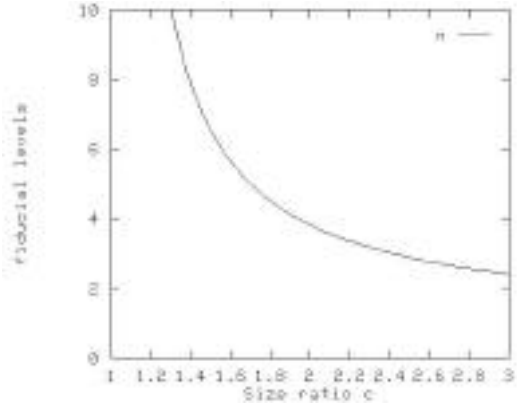



Fig. 6 - Fiducial levels required for an example working range

With the camera at $Z_{far,1}$ from a level $n$ fiducial, the major axis length of the level *1*, *2*, and $j$ rings in an input image are

$$d_{far,1} = \frac{2w}{c-1} c \geq d_{far}$$

$$d_2 = \frac{2w}{c-1} c^2 \leq d_{near,2} \leq d_{near}$$

$$d_j = \frac{2w}{c-1} c^j, (1 \leq j \leq n)$$

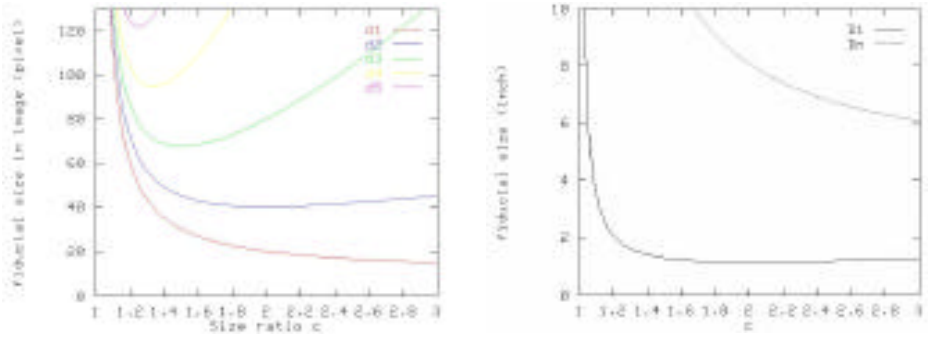
The diameters of the level *1* and *n* fiducials are

$$D_n = \frac{Z_{far}}{f} d_{far} = \frac{Z_{far}}{f} \frac{2w}{c-1} c$$

$$D_1 = \frac{Z_{near}}{f} d_{near} \geq \frac{Z_{near}}{f} \frac{2w}{c-1} c^2$$

$$= c^{1-n} D_n = \frac{Z_{far}}{f} \frac{2w}{c-1} c^{2-n}$$

Figure 7 shows the major axis lengths of some fiducial levels, and the minimum and maximum fiducial sizes as a function of *c*.

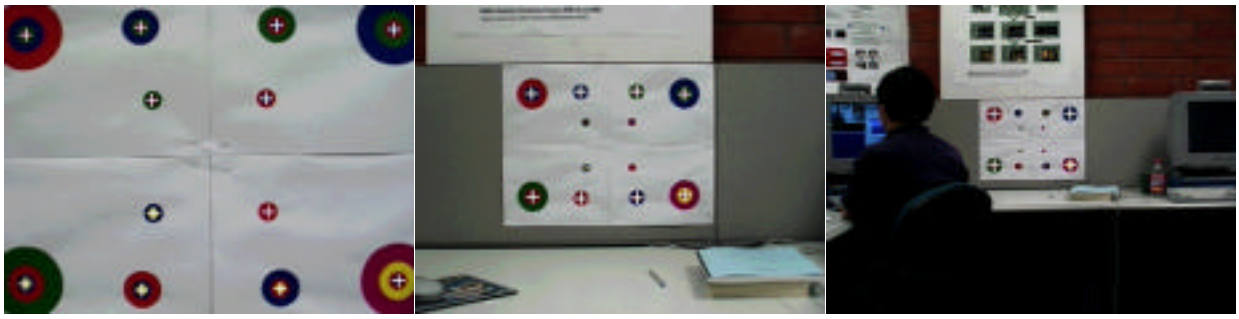



(a) Fiducial sizes in an image (b) Real fiducial sizes

Fig.7 – Image and actual fiducial sizes as a function of *c*

Additional design parameters, such as fiducial spacing constraints are detailed in [Cho98]. An example design is given for tracking in a 20x20 foot room. The camera $FOV_u$=41.7° and $FOV_v$=32°. Let the minimum detectable ring width *w* be five pixels. The closest operating distance is approximately arms-length (two feet) and the farthest distance is the corner-to-corner length $20\sqrt{2}$ feet. The required number of fiducial levels is $\mathbf{log}(20\sqrt{2} / 2) / \mathbf{log}2 \approx 4$. The fiducial sizes in each level are $D_1$ = 1.01", $D_2$ = 2.03", $D_3$ = 4.06", and $D_4$ = 8.1". The inter-fiducial distances are $L_1$ = 5.6", $L_2$ = 11.2", $L_3$ = 22.3", and $L_4$ = 44.5".

A laboratory demonstration uses an SGI Indy (MIPS4400@200MHz with 24-bit graphics) and a SONY DXC-151A color-video camera with S-video output digitized at 640x480 resolution. The lens produces a 31.4° horizontal and 24.37° vertical FOV. A three-level proportional-width fiducial set with six colors (red, green, blue, yellow, cyan, and magenta) is printed on a laser printer. The diameters of the fiducials at each level are: first level = 0.8", second level = 1.6", and third level = 3.2". Rings of 20 – 50 pixels in diameter are detected. The detection range of the first level fiducial is 1.5' - 3.7', the second level 3.0' – 7.4', the third level 5.9 – 14.8'. The combined system detection range



(a) Distance 3 feet - All three levels of fiducials are detected

(b) Distance 6 feet - The second and third level fiducials are detected

(c) Distance 12 feet – Only the third level fiducials are detected

| Fiducial level | Diameter (inch) | Theoretical tracking range (feet) | Distance (feet) | Frame rate (FPS) |
|---|---|---|---|---|
| First level | 0.8 | 1.5 – 3.7 | 3, (a) | 6.3 |
| Second level | 1.6 | 3.0 – 7.4 | 6, (b) | 7.3 |
| Third level | 3.2 | 5.9 – 14.8 | 12, (c) | 8.1 |

Fig. 8 - Detection results. The detected fiducials have a white cross hair at their center

is 1.5' – 14.8'. Figure 8 shows three images of detection results at varied distances from the three level fiducials. Detected fiducials are marked with a white cross-hair at their center. The system performance depends on the number and size of potential fiducials in the image. There is no prediction of fiducial positions, the whole image is processed every frame. The refresh rates are 6.3 frames per second (FPS) at Fig 8 (a), 7.3 FPS at Fig 8 (b), 8.1 FPS at Fig 8 (c).

### 2.3. Pose Estimation

Three of more detected fiducials facilitate pose calculation. Three points provide up to four possible solutions [Fisc81] [Hara94] [Shar97] [Linn88], four non-planar points provide an analytic solution [Hora89]. Recursive filters [Mend95] [Welc97] [Broi90] estimate pose from one or more points sensed at any instant in time. Projective invariants provide a relative projection for five points under full perspective [Ueno95] or four points under weak-perspective [Kutu96]. Model-based methods include [Toma90] [Nato95]. We started with the three-point approach in [Fisc81] and we are currently developing robust recursive filters that use all visible features with position confidence values.

### 2.4. Indoor AR Applications

The combination of feature detection and pose estimation provides the basic capability for vision-tracked AR applications [Caud92] [Fein93] [Shar97]. Human factors and cognitive issues for AR manufacturing and maintenance applications are detailed in [Neum98a]. Figure 9 illustrates a sequence of AR images from a simulated maintenance application. Annotation alignment with the structure is based on detecting dot fiducials and computing camera pose as described above. This example illustrates the simplicity and minimal intrusion offered by sticker fiducials and the flexibility of the approach. The software library developed for this application supports our laboratory research as well as continued experimental application development.

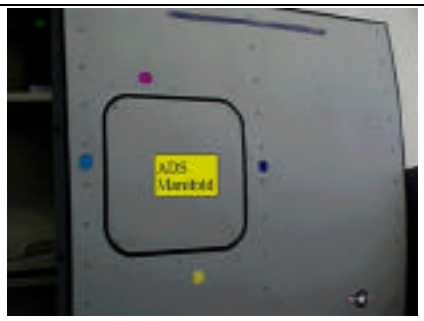
## 2.5. Extendible Tracking

A drawback of tracking from fiducials is the limited range of camera viewpoints from which the fiducials are visible. Camera pan or zoom can quickly move fiducials out of view, resulting in a loss of camera tracking. Allowing users to interactively place new fiducials in the scene, when and where needed, can extend the range of tracked camera positions. The new fiducials are calibrated from the initial fiducials [Neum98] by using recursive filters that estimate the feature positions as the system is used. Once calibrated to sufficient accuracy and confidence, the new fiducials can support pose calculation.

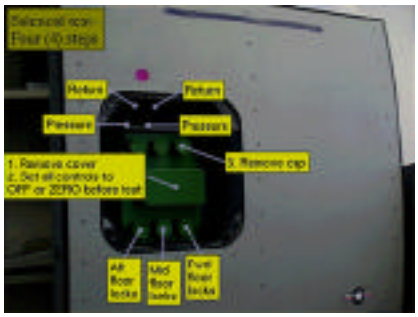
In [Neum98], two recursive filters are used to estimate the 3D positions of new fiducials. The EKF (Extended Kalman Filter) has the 3D position of a new fiducial as its state. The equations for the filter are simple and fast to compute. The EKF is composed of a *predictor* (time update) and *corrector* (measurement update). The state prediction is simply the prior state since the scene is assumed to be rigid and static.

$$\hat{x}_k^- = \hat{x}_{k-1}$$
$$P_k^- = P_{k-1} + Q$$
$$\hat{z}_k = \vec{h}(\hat{x}_k^-, c_k, p_c)$$

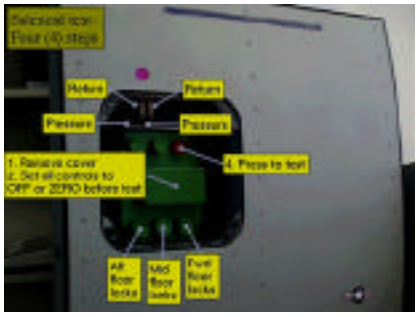
The *corrector* equations correct the predicted state value $\hat{x}_k^-$ based on the residual difference between the actual measurement $z_k$ and the measurement estimate $\hat{z}_k$. The Jacobian matrix linearizes the measurement function.



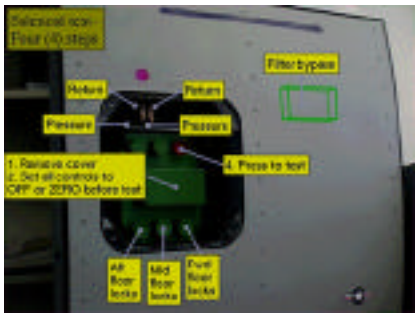

(A) Identification of ADS manifold access panel




(B) Recognition of open panel causes identification of components and instruction to remove cap

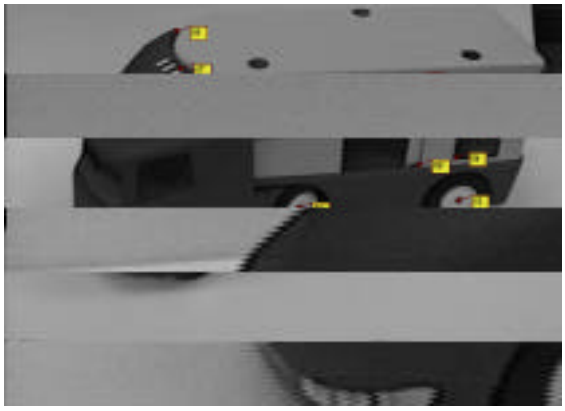



(C) Recognition of cap removal causes new instruction to press button

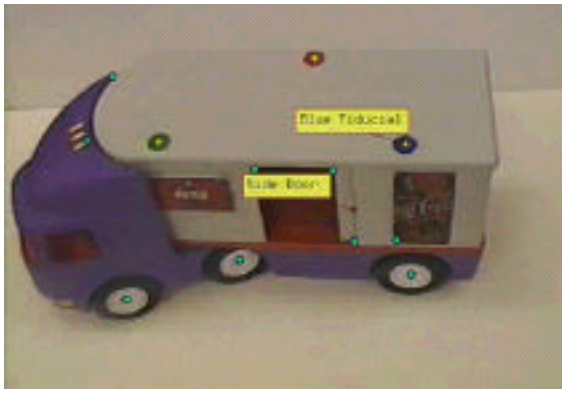



(D) Additional information about position of occluded filter bypass structure in response to user's query

Fig. 9 - Sequence of automatic instructions and information for maintenance task



(a) Tags show positions of automatically detected, tracked, and calibrated natural features.




(b) Initial image with fiducial camera tracking.


(c) Frame 295 with camera tracking based on tracked and calibrated natural features

Fig. 11 – The sequence starts by tracking camera pose from fiducials (b), while natural features are automatically detected and calibrated. Note the annotation indicating the blue fiducial and the side door of the truck. As the camera drops low to the ground at the end of the sequence (c), the fiducials are no longer usable for tracking since their aspect is extreme, and natural features automatically support continued tracking.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}$$
$$\tilde{z}_k = z_k - \hat{z}_k$$
$$\hat{x}_k = \hat{x}_k^- + K_k \tilde{z}_k$$
$$P_k = (I - K_k H_k) P_k^-$$

More detail about Kalman filters can be found in references such as [Mend95, Welc97, Broi90].

The RAC (Recursive Average of Covariances) filter projects the screen measurement into the 3D space occupied by the fiducials. Both filters are stable in practice. The EKF has optimal characteristics under certain conditions [Broi90], however the RAC filter gives comparable results, and it is simpler, operating completely in 3D-world space with 3D lines as measurements. The RAC approach eliminates the Jacobian matrix linearization required in the EKF.

Extendable tracking can make use of fiducials as described above or *natural* scene features such as corners or textures. The latter approach requires methods for natural feature (NF) detection and tracking (correspondence) in the 2D-image sequences produced by a moving camera. A closed-loop NF tracking method [Neum98b] is described in the next section.

Figure 10 illustrates how extendible tracking is achieved with NF tracking in combination with fiducial tracking and new point calibration. Newly detected natural features are tracked in consecutive images. Feature image coordinates are input to a recursive filter that estimates the 3D feature positions. These 3D positions and their corresponding image coordinates are used to track camera pose in the event that any or all of the fiducials become occluded or undetectable.

Experiments with NF tracking are done in off-line but automatically (no user intervention). Natural feature tracking is about an order of magnitude too slow (~1Hz) for real-time applications. Optimizations, dedicated hardware, and DSP processor implementations of the algorithms may lead to real-time implementations. The sequence shown in Figure 11 demonstrates the automatic detection and calibration of natural features during camera motion, and the automatic use of the natural features to perform camera tracking when fiducials become obscured.

## 3. Outdoor Tracking

The challenge for AR in outdoor settings is to track without any modification or preparation of the environment. Visual sensing is still useful, but there are often no recognizable or predictable targets to support tracking. Whatever type of features are in the environment, they must be detected and tracked (corresponded) in 2D image sequences before any camera motion (tracking) is computed. Corner features or points can be detected in many outdoor scenes, however, tracking them is difficult since corners and points are locally similar to each other. Information about the arrangement of groups of features is useful for establishing correspondences. Region tracking can provide both this neighborhood context to aid tracking, as well as a motion prediction for the individual features that helps minimize the search for the best feature correspondence.

### 3.1. Closed-Loop Feature Tracking

A closed-loop approach provides accurate and robust natural feature detection and tracking in natural environments [Neum98b]. The system integrates three main motion-analysis functions, feature selection, motion tracking, and estimate verification, in a closed-loop cooperative manner to track through complex natural imaging conditions. Point features and region features are selected, tracked, and evaluated for their suitability and reliability for computing motions. Continued tracking of either feature type is dependent on the tracking confidence $C$ that is derived from a suitability metric ($\lambda$) and the confidence of tracking ($\delta$).

$$C = k_1\lambda + k_2\delta$$

where $k_i$ is a weighting coefficient for each component.

Region tracking is a differential-based local optical flow estimation [Anan89] [Ball81] [Beau95] [Berg87] [Horn84] [Hild84]. A multi-scale estimation strategy iteratively fits region and point motion estimates to an affine motion model until they agree. The three iterative components (Fig. 12) of the method are image warping, motion residual estimation, and motion refinement. The method operates independently on selected regions of the image. Every region motion estimate is fit to an affine model. A verification and evaluation measures the

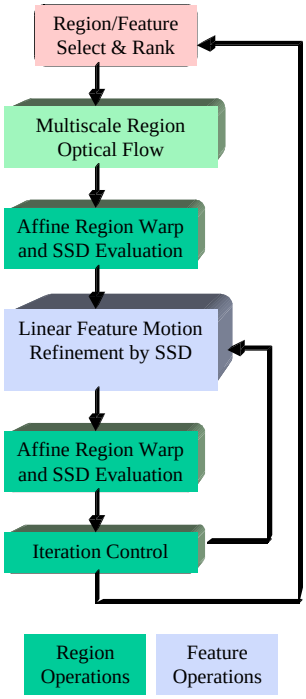



Fig. 12 – Feedback from tracking evaluation controls the iterative refinement of motion estimates within each region.

confidence of the estimation and the model fit. If the estimate error is large, an iterative refinement is done until the error converges or the region is discarded as unreliable for tracking.

An affine model of region motion accounts for the geometric distortions that occur with large view variations and long sequence tracking. A simple translation models point feature motion. These models are the basis of the motion verification and evaluation processing. In regions, the optical-flow motion estimate determines the parameters of an affine motion model that specifies a warp of the region into an confidence evaluation frame $(x_c,y_c)^T$.

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} v_2 x_{t_0} + v_3 y_{t_0} + v_1 \\ v_4 x_{t_0} + v_5 y_{t_0} + v_6 \end{bmatrix}$$

The confidence frame is compared to the true target image to obtain a measure of tracking error

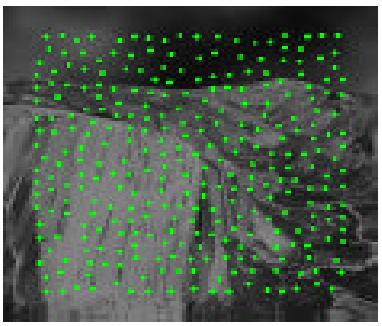
$$\varepsilon = \frac{\left\| R_t(\mathbf{x},t) - R_c(\mathbf{x},t) \right\|^2}{\max\{\left\| R_t(\mathbf{x},t) \right\|^2, \left\| R_c(\mathbf{x},t) \right\|^2\}}$$

where $R_t(\mathbf{x},t)$ and $R_c(\mathbf{x},t)$ are the true target frame and confidence evaluation frame, respectively. The affine parameters predict the translation of point features within a region. The predictions are refined by a local search for a correlation peak. The refined point motions determine new parameters for their region motion model. The new parameters determine a new confidence frame and a new assessment of tracking accuracy. The process of region and point motion modeling and accuracy assessment is iterated until the estimates converge or features are discarded.

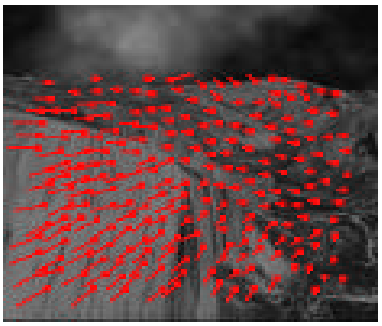
Tracking accuracy feedback is an essential component of the tracking system. As described above, the error information is used by the tracking module for motion correction. The error also used in the feature detection module for continuous feature re-evaluation. Re-evaluation keeps the system working in an "optimum" state by automatically selecting and maintaining only the most reliable features. The tracking accuracy feedback (or tracking confidence) is defined as

$$\delta = \frac{1}{1+\varepsilon}$$

The closed-loop stabilization of the tracking system is inspired by the use of feedback for correcting errors in non-linear control system. The process makes it possible to discriminate between good and poor estimation features, and maximizes the quality of the final motion estimation.



(a) detected tracking features



(b) estimated motion field

Fig. 13 - Synthetic image sequence (Yosemite-Fly-Through ) for performance evaluation

## Accuracy Comparison

Our approach compares favorably to other published methods. Figure 13 is from a synthetic image sequence. Table 2 gives numerical comparisons of the motion errors for this sequence that has camera moving along its view axis towards the mountain and valley. The camera motion generates diverging motion flow around the upper right of the mountain, producing one pixel per-frame translation motion in the cloud area and about four pixels per-frame of motion in the lower-left area. For this test, the original image size is (256x256).

| Technique | Average Angle Error | Standard Deviation |
|---|---|---|
| Horn and Schunck | 11.26 | 16.41 |
| Lucas and Kanade | 4.10 | 9.58 |
| Anandan | 15.84 | 13.46 |
| Fleet and Jepson | 4.29 | 11.24 |
| Closed-loop approach | 2.84 | 7.69 |

Table 2 - Performance comparison for various optical flow approaches with synthetic data sequence

An example of tracking natural features in an outdoor scene is shown in Figure 14 where the camera pans and translates simultaneously. The tracking method automatically chooses the "best" features, and these are highlighted with motion vectors in Figure 14.

Fig. 14 - Tracking result for an outdoor natural scene

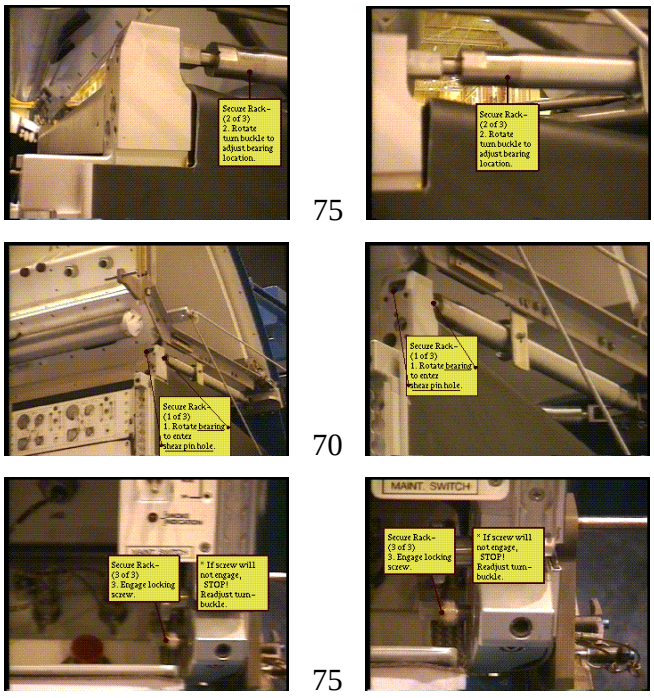
## Angle Error Measure

The angle error measure computed for Table 2 treats image velocity as a spatio-temporal vector $\mathbf{v} = (u, v, 1)$ in units of (pixel, pixel, frame). The angular error between the correct velocity $\mathbf{v}_c$ and the estimate $\mathbf{v}_e$ is

$$Error_{angle} = \arccos(\mathbf{v}_c \ \ \mathbf{v}_e),$$

where $\mathbf{v}_i = \dfrac{(u, v, 1)^T}{\sqrt{u + v + 1}}$.

This angle error measure represents large and small velocities without the amplifications inherent in a relative measure of small vector differences. The measure has potential bias however; directional errors for




(a) key-frame # of tracked frames (b) end frame

Fig. 15 - Direct scene annotation (a) initial frames used to interactively place annotations, (b) later frames in the same sequences showing the automatic tracking of the selected features after the indicated number of frames of hand-held camera motion.

small velocities do not give as large an angular error as similar directional errors with large velocities. For this reason, we also used the RMS measure

$$Error_{rms} = \sqrt{\frac{\sum_{\mathbf{x}\in\Omega} (I_c(\mathbf{x},t) - I_e(\mathbf{x},t))^2}{MN}}$$

where $I_c(\mathbf{x},t)$ is a size $M \leftrightarrow N$ region of a real image sequence at time $t$, and $I_e(\mathbf{x},t)$ is the reconstructed region based on the estimated motion field. Note that this error measure is similar to the motion residual measure we use to assess tracking accuracy.

**Direct Image Annotation**

Robust natural feature tracking enables the direct application of annotation to image sequences. Figure 15 illustrates an application where annotations are interactively placed in an initial frame. The tracking system automatically aligns the annotation in the remaining frames of the sequence. This example portrays information that guides astronauts through a tedious equipment installation.

### 3.2. Hybrid Inertial and Video Tracking

Vision tracking is computationally demanding and susceptible to occlusion or numerical instability. Automatic feature calibration (or extendible tracking) addresses this to some degree, however inertial sensors provide a compelling option. Inertial sensors are completely passive, requiring no external devices or targets, however, their drift rates in portable strapdown configurations are too great for practical use by themselves [Brit71] [Titt97] [Foxl96]. A hybrid approach, integrating inertial and vision-based technologies, can exploit the complementary nature of the two technologies and compensate for their respective weaknesses [Foxl98].

The basic principles behind inertial sensors for determining orientation and position rest on Newton's laws. Accelerometers measure linear acceleration vectors with respect to the inertial reference frame. In order to subtract the acceleration component due to gravity, the orientation of the linear accelerometers must be accurately determined. Rate gyros sense changes in orientation. A time integration of the gyro outputs computes the orientation changes. The integration of gyro signals and errors gives rise to a linear growth in orientation error. Correction techniques may include magnetic compass measurements, however; compass signals are also noisy and especially subject to errors induced by ferrous materials. Indoor and urban compass data are unreliable. Vision-based corrections, on the other hand, may generalize to a wide range of environments. Since accelerometers require orientation for subtracting the gravity vector, we start our inertial sensor investigations with a focus on rate gyros.

**Error Sensitivity of Inertial Tracking**

We experimented with a three-degree of freedom (3DOF) orientation tracker produced by InterSense (Model IS-300). This device incorporates three orthogonal gyroscopes to sense angular rates of rotation along its three perpendicular axes. It also has sensors for the gravity (down) vector and a compass [Foxl92] to compensate for gyro drift. The measured angular rates are integrated to obtain the three orientation measurements (Yaw, Pitch, and Roll). This system is specified as achieving approximately 1° RMS static orientation accuracy and 3° RMS dynamic accuracy, with 150Hz maximum update rate.

Although adequate for interactive applications in virtual reality, this accuracy is inadequate for AR tracking. To demonstrate this, map the specified error into the 2D image domain.

Let ( $f_x$ , $f_y$ ) be the effective horizontal and vertical focal lengths of a video camera (in pixels), and ( $L_x$ , $L_y$ ) represent the horizontal and vertical image resolutions, respectively. The field-of-view (FOV) of the camera is calculated as

$$\theta_x = 2\tan^{-1}\left(\frac{L_x}{2f_x}\right)$$

$$\theta_y = 2\tan^{-1}\left(\frac{L_y}{2f_y}\right)$$

If pixels sample the rotation angles uniformly (Yaw and Pitch), the ratio of image pixel motion to the rotation angles (pixel/degree) is given as

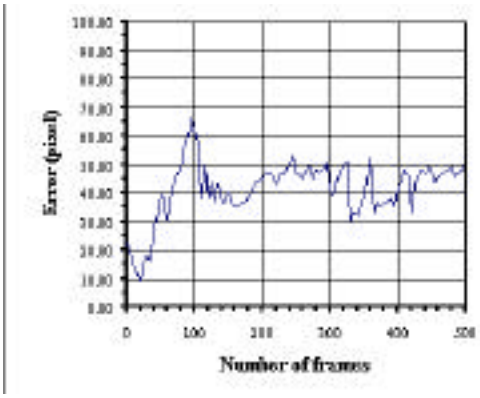
$$L_x/\theta_x = \frac{L_x}{2\tan^{-1}\left(L_x/2f_x\right)}$$

$$L_y/\theta_y = \frac{L_y}{2\tan^{-1}\left(L_y/2f_y\right)}$$

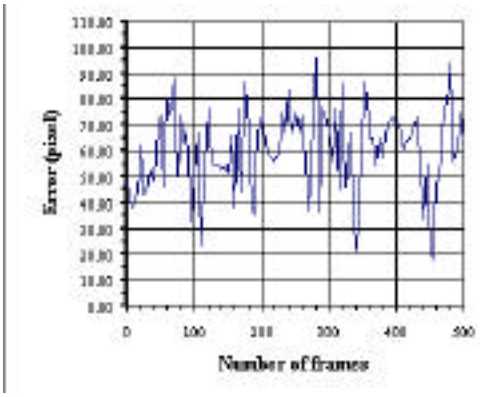
To illustrate a concrete example of this relationship, consider the Sony XP-999 CCD video camera. Through calibration, we determine the effective horizontal and vertical focal lengths as $f_x$ =614.059 pixels, and $f_y$ =608.094 pixels, when digitized at a 640×480 image resolution. The ratios are $L_x/\theta_x$ =11.625 pixel/degree, and $L_y/\theta_y$ =11.143 pixel/degree. That is, each degree of orientation-angle error results in about 11-pixels of image alignment error. In actual use, the error of the inertial tracker may become larger than the one-degree specified. Increasing the FOV of the camera with a wide-angle lens reduces the pixel error proportionately, however wide-angle lenses produce significant radial distortions that also contribute to pixel error [Baju92].

Figure 16 illustrates the measured dynamic alignment error obtained with the inertial gyro. In this experiment, the gyro sensor is attached to a video camera to continually report its orientation. There is no measure of ground-truth absolute pose of the sensor/camera, rather the visual feature motions are tracked to evaluate the gyro sensor accuracy relative to the image. By back-projecting the 3D orientation changes reported by the inertial sensor, the gyro motion estimates can be compared with the observed feature motions in the image plane. Changes in the image-space distances are proportional to any errors accumulated by the inertial system. The error measure is appropriate since the ultimate metric of any augmented reality is the perceived image.

As the camera pans around its azimuth axis, video images are captured in real time, and the inertial tracker continually reports the camera orientation. At the start of each test, ten distinct visual features are selected interactively. These features are automatically tracked in subsequent frames, and their positions are compared to the back projected positions computed from the gyro motion data. The average distance between the vision-tracked feature positions and their corresponding backprojected positions is the accuracy metric. Two 500 frame sequences, a far-view (>100 feet) and near-view* scene, are used for the test. Figure 16 illustrates the average error distributions for the two scenes, and it clearly shows the dynamic drifts between the gyro data and tracked features.




(a) Far-view scene




(b) Near-view scene

Fig 16 – Average pixel differences between tracked features and backprojected features for (a) distant and (b) near scenes.

## Hybrid Tracking Approach

Our prototype hybrid system fuses inertial orientation (3DOF) data with vision feature tracking to improve vision robustness and correct inertial drift. We treat the fusion as a 2D image stabilization problem. Approximate 2D feature-motion is derived from the inertial data, and vision tracking corrects and refines these estimates in the image domain. Furthermore, the inertial data is a prediction that reduces the correspondence search space and provides tolerance to vision tracking interruptions.

## Camera Model and Coordinates

A CCD video camera has a rigidly mounted 3DOF inertial sensor. There are four principal coordinate systems, as illustrated in Figure 17, the world coordinate system $\mathbf{W}:(x_w, y_w, z_z)$, the camera-centered coordinate system $\mathbf{C}:(x_c, y_c, z_c)$, the inertial-centered coordinate system

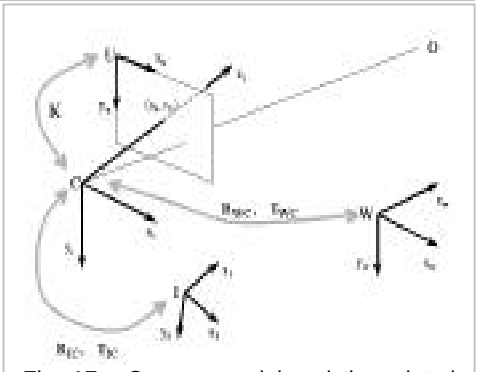


Fig. 17 - Camera model and the related coordinate systems of the hybrid system

* We only consider pure rotation of the camera. Although we carefully pan the camera to avoid translations, a minor translation is injected by the offset between the rotation axis and the optical center of the camera. For completeness, we consider both a far-view scene with feature ranges of over 100 feet and a near-view (12x12 foot office) scene that is more sensitive to minor translation.

$\mathbf{I}:(x_I, y_I, z_I)$, and the 2D image coordinate system $\mathbf{U}:(x_u, y_u)$.

A pinhole camera models the imaging process. The origin of **C** is at the projection center of camera. The transformation from **W** to **C** is

$$\mathbf{W}:\rightarrow \mathbf{C}: \quad \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \left[\mathbf{R}_{wc} \middle| -\mathbf{R}_{wc}\mathbf{T}_{wc}\right] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

where the rotation matrix $\mathbf{R}_{wc}$ and the translation vector $\mathbf{T}_{wc}$ characterize the orientation and position of the camera with respect to the world coordinate frame. Under perspective projection, the transformation from **W** to **U** is

$$\mathbf{W}:\rightarrow \mathbf{U}: \quad \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = [\mathbf{K}] \left[\mathbf{R}_{wc} \middle| -\mathbf{R}_{wc}\mathbf{T}_{wc}\right] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

where the matrix **K**

$$\mathbf{K} = \begin{bmatrix} \alpha_x f & 0 & u_0 \\ 0 & \alpha_y f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

contains the *intrinsic* parameters of the camera[*], $f$ is the focal length of camera, $\alpha_x, \alpha_y$ are the horizontal and vertical pixel sizes on the imaging plane, and $(u_0, v_0)$ is the projection of camera center (principal point) on the image plane. The intrinsic parameters are calibrated offline.

Camera orientation changes are reported by the inertial tracker, so the transformation between the **C** and **I** is needed to relate the inertial and camera motions. For rotation $\mathbf{R}_{Ic}$ and translation $\mathbf{T}_{Ic}$ the transformation is

$$\mathbf{I}:\rightarrow \mathbf{C}: \quad \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \left[\mathbf{R}_{Ic}\right] \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} + \left[\mathbf{T}_{Ic}\right]$$

Since we only use a 3DOF orientation tracker, only the rotation transformation needs to be determined. An automatic calibration method is detailed below.

**Static Calibration**

Camera calibration determines the intrinsic parameters **K** and the lens distortion parameters. We use the method described in [Neum96]. A planar target with a known grid pattern is imaged at measured offsets along the viewing direction. The intrinsic parameters and coefficients of radial lens distortion are computed by an iterative least-squares estimation. These parameters remain constant during our tracking experiments.

[*] For simplicity we omitted the lens distortion parameters from the equation.

The transformation between the inertial and the camera coordinate systems relates the inertial data to the camera motion, and hence to the image feature motions. Measuring this transformation can be difficult, especially with optical see-through display systems. We describe a motion-based calibration, as opposed to the boresight techniques presented in [Azum94]. As discussed above, only the rotation component of the transformation is determined.

Rewriting the relationship between the inertial tracker frame and the camera coordinate frame as

$$\mathbf{x_C} = \left[\mathbf{R}_{Ic}\right]\mathbf{x_I} + \left[\mathbf{T}_{Ic}\right]$$

If we consider the vector $\mathbf{x_C} = [x_c, y_c, z_c]^T$ and $\mathbf{x_I} = [x_I, y_I, z_I]^T$ as directions to a point in a scene relative to **C** and **I**, the rotation motion relationship between the two coordinates can be derived

$$\varpi_{\mathbf{C}} = \left[\mathbf{R}_{Ic}\right]\omega_{\mathbf{I}}$$

$\omega_{\mathbf{C}} = [\omega_{Cx}, \omega_{Cy}, \omega_{Cz}]^T$ and $\omega_{\mathbf{I}} = [\omega_{Ix}, \omega_{Iy}, \omega_{Iz}]^T$ denote the angular velocity of scene points, relative to the camera coordinate frame and the inertial coordinate frame, respectively.

The angular motion $\omega_{\mathbf{I}}$, relative to the inertial coordinate system, is obtained from the inertial tracker output. We need to compute the camera's angular velocity $\omega_{\mathbf{C}}$ in some way, in order to determine the transformation matrix $\mathbf{R}_{Ic}$. General camera motion can be decomposed into a linear translation $\mathbf{V_C} = [V_{Cx,}, V_{Cy}, V_{Cz}]^T$ and an angular motion $\omega_{\mathbf{C}} = [\omega_{Cx}, \omega_{Cy}, \omega_{Cz}]^T$. Under perspective projection, the 2D-image motion resulting from camera motion can be written as

$$\dot{x}_u = \frac{-f V_{Cx} + x_u V_{Cz}}{z_{\mathbf{C}}} + \frac{x_u y_u}{f}\omega_{Cx} - f\left(1 + \frac{x_u^2}{f^2}\right)\omega_{Cy} + y_u \omega_{Cz}$$

$$\dot{y}_u = \frac{-f V_{Cy} + y_u V_{Cz}}{z_{\mathbf{C}}} + f\left(1 + \frac{y_u^2}{f^2}\right)\omega_{Cx} - \frac{x_u y_u}{f}\omega_{Cy} + x_u \omega_{Cz}$$

where $(\dot{x}_u, \dot{y}_u)$ denotes the image velocity of point $(x_u, y_u)$ in the image plane, $z_{\mathbf{C}}$ is the range to that point, and $f$ is the focal length of camera. Eliminating the translation term and substituting for $\omega_{\mathbf{C}}$, we have $\dot{\mathbf{x}}_u = \Lambda\left[\mathbf{R}_{Ic}\right]\omega_{\mathbf{I}}$, where

$$\Lambda = \begin{bmatrix} \frac{x_u y_u}{f} & -f\left(1 + \frac{x_u^2}{f^2}\right) & y_u \\ f\left(1 + \frac{y_u^2}{f^2}\right) & -\frac{x_u y_u}{f} & -x_u \end{bmatrix}$$

In words, given knowledge of the internal camera parameters, the inertial tracking data $\omega_{\mathbf{I}}$, and the related 2D motions $\left[\dot{x}_u, \dot{y}_y\right]$ of (at least 5) image features, the transformation $\mathbf{R}_{Ic}$ between the camera and the inertial coordinate systems can be determined.

**Dynamic Registration**

Static registration establishes an initial calibration, however the inertial system accumulates drift and errors with motion. The distribution of drift and error (Fig. 16) is difficult to model for analytic correction. Our strategy is to minimize the tracking error in the image plane, relative to the visually-perceived image.

Suppose $N$ points are annotated in the scene. Their projections in the image are $(x_i, y_i)$, $i = 1,2 \cdots N$. Our goal is to automatically track these features as the camera moves in the subsequent frames. Inertial data provides a tracking prediction, and vision tracking provides a correction.

Let $\omega_C = [\omega_{Cx}, \omega_{Cy}, \omega_{Cz}]^T$ be the camera rotation from frame $I(\mathbf{x}, t-1)$ to frame $I(\mathbf{x}, t)$. For the scene points $O_i$, their 2D positions in the image frame $t-1$ are $\mathbf{x}_{it-1} = [x_{it-1}, y_{it-1}]^T$. The positions of these points in the frame $t$, due to the related motion (rotation) between the camera and the scene, can be estimated

$$\mathbf{x}_{it} = \mathbf{x}_{it-1} + \Delta \mathbf{x}_{it}$$

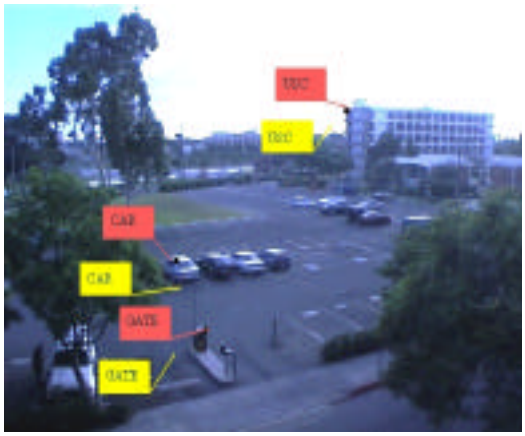
$$\Delta \mathbf{x}_{it} = \Lambda \, \omega_C$$

where $i = 1,2 \cdots N$, and $\Lambda$ is determined as above.

The predicted image positions are refined by local searches for the true features. A normalized correlation is used as the feature matching metric. The feature located at $\mathbf{x}_{it-1}$ in frame $t-1$ is correlated with frame $t$ around the location $\mathbf{x}_{it}$ predicated by the inertial tracker. The result of the search forms a correlation surface, and the location of the peak value of the surface is the correct (best) match. Correlation is defined as

$$\varepsilon = \frac{(2n+1)^2 \mu_{t-1t} - \mu_{t-1}\mu_t}{\sqrt{(2n+1)^2 \sigma_{t-1} - \mu_{t-1}^2}\sqrt{(2n+1)^2 \sigma_t - \mu_t^2}}$$

To minimize the effect of pixel position quantization, 2D bilinear interpolation refines the estimate of the peak to subpixel accuracy.



(a) far-view

(b) near-view

Fig. 18 - Images from two video sequences showing vision (red) and gyro (yellow) tracking results

## Experiment

Figure 18 shows two images from far-view and near-view video sequences. In these frames, black dots identify the feature points that we track and annotate. The yellow boxes are annotation text banners positioned only from inertial data, while the red boxes denote the vision-corrected positions. The resolution of the images is 640x480.

## Inertial-Only Tracking

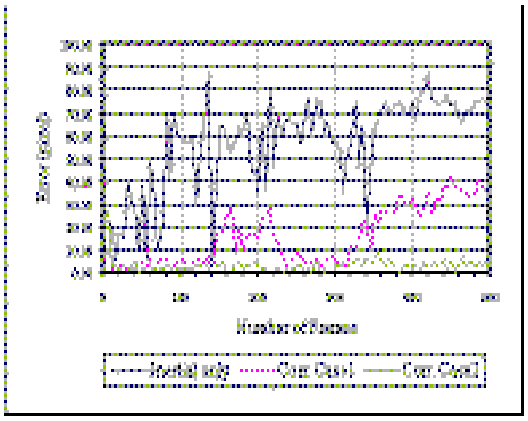
This test uses only inertial data for tracking. Ten distinct features are manually selected in the initial frame to establish visual reference points. The selected features are backprojected in each frame based on the camera orientation change reported by the inertial tracker. The average differences between the backprojected image positions and the observed (vison-tracked) feature positions are the measure of tracking accuracy in each frame. Figure 19 shows the average error distributions for the two scenes confirming that substantial errors occur.

## Hybrid Inertial-Vision: case 1

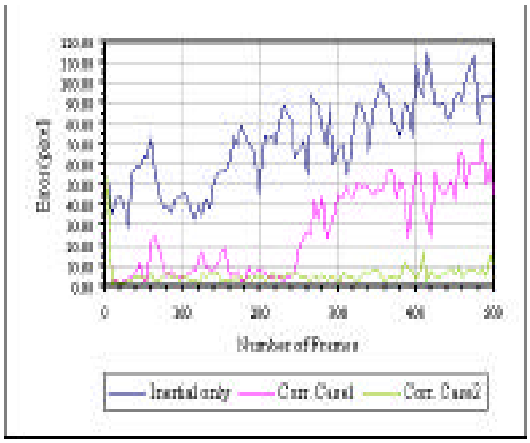
This test performs inertial tracking and vision-based correction of the integrated gyro error. The predication of 2D image motion is based on the integrated inertial orientation. This approach allows the inertial drift to accumulate, unaffected by any vision-based corrections. This simulates the effect of prolonged occlusion or tracking loss in the vision system. Figure 19 illustrates the results for both test scenes.

## Hybrid Inertial-Vision: case 2

An alternative correction approach is incremental. Each vision correction results in an adjustment of the gyro state, consequently, the gyro error accumulation (assuming ideal corrections) is limited to periods between corrections. (This is similar to the "ZUPT'ing" method used for linear motion sensing.) The reduced drift integration periods result in better prediction and better registration as illustrated in figure 19. A drawback of this approach is the possibility that a spurious correction error produces a lingering bias in the result.



(a) far-view sequence



(b) near-view sequence

Fig. 19 – Hybrid alignment error for (a) far-view and (b) near-view scenes with inertial only, hybrid case1, and hybrid case2 methods.

## 4. Summary and Conclusions

We described our views and strategies to track in (almost) natural environments. Robust fiducial detection, scalable fiducials, and extendible tracking enable vision tracking for indoor settings. The closed-loop approach to natural feature tracking provides the robust long-sequence 2D correspondences needed for natural settings. A fusion of inertial gyro and vision capabilities produce stable and robust orientation measures.

Much remains undone. Other sensors and representations of data can be brought to bear on this critical problem of natural environment tracking.

## 5. Acknowledgments

Portions of this work are supported by the Defense Advanced Research Project Agency (DARPA) "Geospatial Registration of Information for Dismounted Soldiers." The Integrated Media Systems Center provides support and facilities. Recognition for much of the work described here goes to the research members of the AR Tracking Group in the Computer Science Department at the University of Southern California, Dr. Suya You, Youngkwan Cho, Jongweon Lee, Jun Park, and Bolan Jian.

## 6. References

[Anan89] P. Anandan. A Computational Framework and an Algorithm for the Measurement of Visual Motion. International Journal of Computer Vision, Vol. 2, pp. 283-310, 1989.

[Azum95] R. Azuma. A Survey of Augmented Reality. SIGGRAPH 95 course #9 notes, August 1995.

[Azum94] R. Azuma and G. Bishop, Improved Static and Dynamic Registration in an Optical See-through HMD, SIGGRAPH 1994, pp. 197-203

[Baju95] M. Bajura and U. Neumann. Dynamic Registration Correction in Augmented Reality Systems. Proc. of IEEE Virtual Reality Annual International Symposium, pp. 189-196, 1995.

[Ball81] D. H. Ballard and C. M. Brown, Computer Vision, Prentice Hall (1981), ISBN 0-13-165316-4

[Beau95] S. S. Beauchemin and J. L. Barron. The Computation of Optical Flow. ACM computing surveys, Vol. 27, No 3, pp. 433-466, 1995.

[Berg87] J. R. Bergen and E. H. Adelson. Hierarchical, Computationally Efficient Motion Estimation Algorithm. J. Opt. Soc. Am. Vol. 4, No. 35, 1987

[Brit71] K. Britting. Inertial Navigation System Analysis. Wiley Interscience, New York, 1971.

[Broi90] T. J. Broida, S. Chandrashekhar, R. Chellappa, "Recursive Estimation from a Monocular Image Sequence," IEEE Transactions on Aerospace and Electronic Systems, Vol. 26, No. 4, pp. 639-655, July 1990

[Caud92] T. P. Caudell and D. M. Mizell, Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes. Proc. of the Hawaii International Conference on Systems Sciences, pp. 659-669, 1992.

[Cho97] Y. Cho, J. Park, and U. Neumann, Fast Color Fiducial Detection and Dynamic Workspace Extension in Video See-through Augmented Reality, Proceedings of the Fifth Pacific Conference on Graphics and Applications, October 1997, pp. 168-177

[Cho98] Y. Cho and U. Neumann, "Multi-ring Color Fiducial Systems and A Detection Method for Scalable Fiducial-Tracking Augmented Reality", submitted to First International Workshop on Augmented Reality, 1998

[Eagl] Eagle Eye, Kinetic Sciences http://www.kinetic.bc.ca/eagle_eye.html

[Fein93] S. Feiner, B. MacIntyre, D. Seligmann. Knowledge-Based Augmented Reality. Communications of the ACM, Vol. 36, No. 7, pp. 52-62, July 1993.

[Fisc81] M.A. Fischler, R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Graphics and Image Processing, Vol. 24, No. 6, 1981, pp. 381-395.

[Flee90] D. J. Fleet and A. D. Jeson. Computation of Component Image Velocity from Local Phase Information. International Journal of Computer Vision, Vol. 5, pp. 77-104, 1990.

[Foxl96] E. Foxlin. Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter. Proc. of IEEE Virtual Reality Annual International Symposium, pp. 184-194, 1996.

[Foxl98] E. Foxlin, M. Harrington, and G. Pfeifer. Constellation: A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications. Prof. of GRAPHICS 98, 1998.

[Ghaz95] M. Ghazisadedy, D. Adamczyk, D. J. Sandlin, R. V. Kenyon, T. A. DeFanti. Ultrasonic Calibration of a Magnetic Tracker in a Virtual Reality Space. Proc. of IEEE Virtual Reality Annual International Symposium pp. 179-188, 1995.

[Hage96] G. D. Hager and P. N. Belhumeur. Real-Time Tracking of Image Regions with Changes in Geometry and Illumination. Proc. of IEEE CVPR, 1996.

[Hara94] R. Haralick, C. Lee, K. Ottenberg, and M. Nolle, Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem, IJCV, Vol. 13, No. 3, 1994, pp. 331-356

[Heal97] C.G. Healey and J.T. Enns, A Perceptual Colour Segmentation Algorithm, Technical Report, University of British Columbia, 1997

[Hild84] E.C. Hildreth. Computation Underlying the Measurement of Visual Motion. Artificial Intelligence, Vol. 23, pp. 309-354, 1984.

[Hora89] R. Horaud, B. Conio, and O. Leboulleux, An Analytic Solution for the Perspective 4-Point Problem, CVGIP 47, 1989, pp. 33-44

[Horn84] B. K. P. Horn and B. G. Schunk. Determining Optical Flow. Artificial Intelligence, Vol. 17, 185-203, 1984.

[Hun95] B.R. Hunt, Superresolution of Images: Algorithms, Principles, Performance, IJIST(6), No. 4, 1995, pp. 297-304

[Kend76] J. Kender, Saturation, hue, and normalized color: Calculation, Digitization Effects, and use, CMU Computer Science Dept., Nov. 1976

[Kim97] D. Kim, S. W. Richards, T. P. Caudell. An Optical Tracker for Augmented Reality and Wearable Computers. Proc. of IEEE Virtual Reality Annual International Symposium, pp. 146-150, 1997.

[Klin97] G. Klinker, K. Ahlers, D. Breem, P. Chevalier, C. Crampton, D. Greer, D. Koller, A. Kramer, E. Rose, M. Tuceryan, R. Whitaker, "Confluence of Computer Vision and Interactive

Graphics for Augmented Reality," Presence: Teleoperator and Virtual Environments, Vol. 6, No. 4, pp. 433-451, August 1997

[Koll97] D. Koller, G. Klinker, E. Rose, D. Breen, R. hitaker, and M. Tuceryan, Real-time Vision-Based Camera Tracking for Augmented Reality Applications, Proceedings of the Symposium on VRST, Lausanne, Switzerland, September, 1997, pp. 87-94

[Kutu96] K. Kutulakos, J. Vallino. Affine Object Representations for Calibration-Free Augmented Reality. Proc. of IEEE Virtual Reality Annual International Symposium, pp. 25-36, 1996.

[Lee98] J. Lee, U. Neumann, "Fuzzy and Rule-Based Fiducial Detection for Vision-based Augmented Reality Tracking", submitted to First International Workshop on Augmented Reality, 1998

[Linn88] S. Linnainmaa, D. Harwood, and L.S. Davis, Pose Determination of a Three-Dimensional Object Using Triangle Pairs, PAMI, Vol. 10, No. 5, September 1988, pp. 634-647

[Luca96] B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. Proc. DARPA IU Workshop, pp. 121-130, 1981.

[Madr96] F. Madritsch, F. Leberl, and M. Gervautz, Camera based Beacon Tracking: Accuracy and Applications, VRST '96, pp.101-108, 1996

[Mell95] J.P. Mellor, Enhanced Reality Visualization in a Surgical Environment, AI Technical Report 1544, 1995

[Mend95] J.M. Mendel, "Lessons in Estimation Theory for Signal Processing, Communications, and Control", Prentice Hall PTR, 1995

[Meye92] K. Meyer, H.L. Applewhite, F.A. Biocca. A Survey of Position Trackers. Presence: Teleoperators and Virtual Environments, Vol. 1, No. 2, pp. 173-200, 1992.

[Nage89] H. H. Nagel. On a Constraint Equation for the Estimation of Displacement Rates in Image Sequences. IEEE Trans. PAME-1, Vol. 1, pp.13-30, 1989.

[Nato95] E. Natonek, Th. Zimmerman, L. Fluckiger, " Model Based Vision as Feedback for Virtual Reality Robotics Environments," Proceedings of VRAIS'95, pp. 110-117

[Neum96] U. Neumann and Y. Cho, A Self-Tracking Augmented Reality System. Proc. of ACM Virtual Reality Software and Technology, pp. 109-115, 1996.

[Neum98] U. Neumann and J. Park. Extendible Object-Centric Tracking for Augmented Reality. Proc. of IEEE Virtual Reality Annual International Symposium, pp.148-155, 1998.

[Neum98a] U. Neumann and A. Majoros. Cognitive, Performance, and Systems Issues for Augmented Reality Applications in Manufacturing and Maintenance. Proc. of IEEE Virtual Reality Annual International Symposium, pp. 4-11, 1998.

[Neum98b] U. Neumann and S. You. Integration of Region Tracking and Optical Flow for Image Motion Estimation. Proc. of IEEE International Conference on Image Processing, 1998.

[Oakl91] J.P. Oakley and R.T. Shann, Efficient Method for Finding the Position of Object Boundaries to Sub-Pixel Precision, IVC(9), 1991, pp. 262-272

[Papo91] A. Papoulis, Probability, Random Variables, and Stochastic Processes Third Edition, McGraw-Hill, New York, 1991

[Prat91] W.K. Pratt, Digital Image Processing, A Wiley-Interscience Publication

[Reki97] J. Rekimoto, "NaviCam: A Magnifying Glass Approach to Augmented Reality," Presence: Teleoperator and Virtual Environments, Vol. 6, No. 4, pp. 399-412, August 1997

[Shar97] R. Sharma, J. Molineros. Computer Vision-Based Augmented Reality for Guiding Manual Assembly. Presence: Teleoperators and Virtual Environments, Vol. 6, No. 3, pp. 292-317, June 1997.

[Sowi93] H. Sowizral, J. Barnes, "Tracking Position and Orientation in a Large Volume," Proceedings of IEEE VRAIS'93, pp. 132-139

[Star97] T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsh, R. Picard, A. Pentland, "Augmented Reality Through Wearable Computing," Presence: Teleoperator and Virtual Environments, Vol. 6, No. 4, pp. 386-398, August 1997

[Stat96] A. State, G. Hirota, D. T. Chen, B. Garrett, M. Livingston. Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. Proc. of SIGGRAPH'96, pp. 429-438, 1996.

[Suth68] I. Sutherland "A Head-Mounted Three-Dimensional Display," Fall Joint Computer Conference, pp. 757-775, 1968

[Titt97] D.H. Tittertion and J. L. Weston. Strapdown inertial navigation technology. IEE Radar, Sonar, Navigation and Avionics Series 5, Peter Peregrinus Ltd. UK 1997.

[Toma90] T. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method. Technical Report, Carnegie Mellon University, Pittsburgh, PA, September 1990.

[Trem97] A. Tremeau, N. Borel, "A Region Growing and Merging Algorithm to Color Segmentation," Pattern Recognition, Vol. 30, No. 7, pp. 1191-1203, 1997

[Tuce95] M. Tuceryan, D. S. Greer, P. T. Whitaker, D. 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, Vol. 1, No. 3, pp. 255-273, September 1995

[Ueno95] M. Uenohara, T. Kanade. Vision-Based Object Registration for Real-Time Image Overlay. Proc. of Computer Vision, Virtual Reality, and Robotics in Medicine, pp. 13-22, 1995.

[Ward92] M. Ward, R. Azuma, R. Bennett, S. Gottschalk, H. Fuchs, "A Demonstrated Optical Tracker with Scalable Work Area for Head-Mounted Display Systems," Proceedings of the 1992 Symposium on Interactive 3D Graphics," pp. 43-52

[Welc97] G. Welch, G. Bishop, "SCAAT: Incremental Tracking with Incomplete Information," Proceedings of Siggraph97, Computer Graphics, pp. 333-344