Automated Initialization for Marker-less Tracking: A Sensor Fusion Approach

Hesam Najafi Nassir Navab Gudrun Klinker
Technische Universität München, Institut für Informatik
Boltzmannstr. 3, 85748 Garching b. Muenchen, Germany
{*najafi, navab, klinker*}*@cs.tum.edu*

Abstract

We introduce a novel sensor fusion approach for automated initialization of marker-less tracking systems. It is not limitated in tracking range and working environment, given a 3D model of the objects or the real scene.

This is achieved based on a statistical analysis and probabilistic estimation of the uncertainties of the tracking sensors. The explicit representation of the error distribution allows the fusion of different sensor data, e.g. of mobile tracking sensors with stationary sensors, in order to estimate the initial pose and improve the registration accuracy.

This methodology was applied to an augmented reality system, using a mobile camera and several stationary tracking sensors, and can be easily extended to the case of anny additional sensors. The initialization consists of an iterative pose estimation and refinement process using both stationary and mobile cameras. Thereby the registration error is minimized in 3D object space rather than in 2D image.

Experimental results show how complex objects can be registered efficiently and accurately to an single initial image.

1. Introduction

Tracking mobile users and objects is a central part of every augmented reality (AR) system. Among many available tracking technologies the vision-based tracking systems are in many application fields the method of choice due to their accuracry and flexibility. Thereby, tracking is known as the pose estimation problem, and means estimating the rigid transformation that relates 2D camera images to 3D geometry. While most of the current vision-based trackers rely on markers, there have been there are a few efforts addressing marker-less tracking in the literature [7, 24, 5, 17, 23, 18]. Because of several reasons marker-less tracking with mobile cameras is still one of the currently most challenging tasks in augmented reality.

A main crucial problem with the current marker-less tracking systems that make them yet not suitable in many industrial applications is the automated initialization, i.e. providing the system automatically with the initial pose of the user's view or camera. This procedure also needs to be done each time the system looses track e.g. due to fast movements or occlusion.

We propose an automated initialization approach for indoor as well as outdoor environments. The initial positional data can be provided by stationary cameras in closed buildings and for instance by GPS for outdoors. The correct initialization is achieved by fusing the data from multiple sensors, e.g. mobile and stationary camras or GPS. In cases where tracking is lost for instance because of large occlusions, the initialization procedure is started automatically. This approach can be easily extended to the case of anny additional sensors.

Our system requires a rough 3D model of the target object or the scence for automatic initialization and tracking. This is not a problem in practice, since in many application a 3D model already exists (e.g. in automotive industry) or can be created automatically or easily interactive by commercially available software like ImageModeler from RealViz, Canoma from Metacreations or Boujoi from 2D3. Furthermore, technological advances in three-dimensional scanning provide accurate devices for automatic model building.

A few approaches propose methods for marker-less tracking based on natural features [7, 24, 13, 17, 23, 5, 18].

Genc et al. [7] proposed a general learning-based framework for feature-based tracking using a single camera. In a two stage process first a set of natural 3D features is learned using an external tracking system (e.g. marker-based). In the second stage the system uses these learned features for tracking as soon as enough stable ones are acquired in the first stage. Their marker-less tracking system needs an initialization that provides a rough estimate of the cameras position and orientation. They make use of an external marker-based tracker for initialization. Once the system looses track it needs to be re-initialized in the same way. They however confirm that such initialization solution is not ac-

cepted by users. In this case the initialization process does not need to be very accurate and in perform real-time. The system is able to converge even with partial or imprecise tracking information for initializating system.

Vacchetti et al. [24, 13] propose an automatic initialization method that relies on a learning stage, wehere a data base of key features is constructed based on a set of keyframes taken during an offline procedure. The key features consist of a 3D point on the object model and a viewpoint invariant local descriptor based on its appearance in the images. The initialization is done by robustly matching feature points in the initial image with the points present in the database based on a similarity measure. A disadvantage of this method is that these local descriptors are sensitive to scale and zooming. Therefore the working space is limited in tracking area that is covered by sufficiently enough key frames.

Satoh et al. [21] use a bird's-eye view camera additional to the mobile camera that constrains the pose estimation problem. Nevertheless, they don't make use of it in the initialization phase. The initial registration is done each time manually by moving the mobile camera closely to a predefined initial pose.

Our intention is to use stationary cameras for non-precise tracking of a user's head and combine the tracking data with those acquired by mobile cameras. Thereby special attention is given to the statistical analysis of the errors in sensors. Specially suitable for this purpose are the networked smart cameras. These cameras are equiped with integrated processors and signal processor chips that can immediately process images formed on the sensor chips. The cameras are thus autonomous, i.e. independent from a computer, and provide their tracking results via wireless network to mobile or stationary PCs where the AR applications are running.

This is designed to be integrated in an ubiquitous tracking environment [25], where different tracking sensors with different modalities are used to build dynamically extendible networks of trackers with high-precision, low-latency requirements. The proposed approach can be extended to any kind of trackers, since the uncertainties of those individual trackers are taken into account.

In this paper we will focus on the automated initialization of a marker-less tracking system. The paper is organized as follows: First the mathematical problem definition is given in section 2.1. In section 2.2 we discuss the related work. Section 3 gives a general overview of our method and explains the details of it in the following subsections. In section 4 we present our experimental results. Conclusions and future work are provided in sections 5.

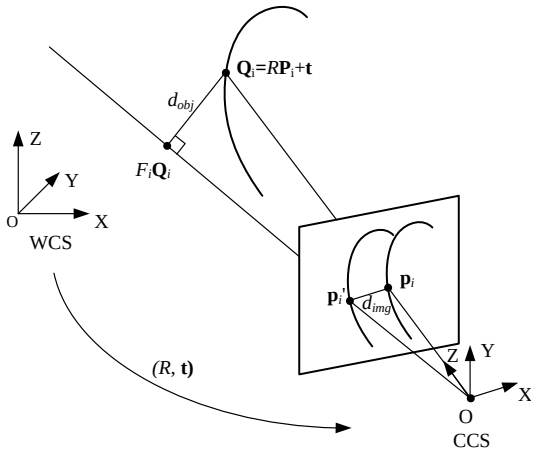

Figure 1. The collinearity errors in object and image space.

2 Background

2.1 Mathematical Problem Definition

Given a set $\{\mathbf{P}_i|0 < i \leq n\}$ of n object points $\mathbf{P}_i = (X_i, Y_i, Z_i)^t$, in the world coordinate system (WCS), the set of corresponding coordinates $\mathbf{Q}_i = (X_i, Y_i, Z_i)^t$ in the camera coordinate system (CCS), are related by a rigid transformation

$$\mathbf{Q}_i = R\mathbf{P}_i + \mathbf{t}$$

where $R = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]^t$ is a 3×3 rotation matrix and $\mathbf{t} = (t_x, t_y, t_z)^t$ is a translation vector.

We choose the projection center of the camera as the origin of the camera coordinate system with the optical axis pointing in the positive z direction. The object points are projected onto the plane with $z = 1$, the *normalized image plane* in the camera coordinare system. We assume that the internal calibration parameters of the camera, e.g. focal length, principal point, lens distortion, etc. are known. The image point $\mathbf{p}_i = (u_i, v_i, 1)^t$ is the perspective proejction of the object point $\mathbf{P}_i$ to the normalized image plane according to the following equation.

$$\mathbf{p}_i = \frac{R\mathbf{P}_i + \mathbf{t}}{\mathbf{r}_3\mathbf{P}_i + t_z}. \quad (1)$$

This equation is called the *collinearity equation* and says that $\mathbf{p}_i$, $\mathbf{Q}_i$ and the projection center of the camera $\mathbf{O}$ are collinear.

However, another way of thinking of collinearity is that the orthogonal projection of $\mathbf{Q}_i$ on the projection ray of $\mathbf{p}_i$ is equal to $\mathbf{Q}_i$ itself [15]. This can be formulated as

$$R\mathbf{P}_i + \mathbf{t} = F_i(R\mathbf{P}_i + \mathbf{t}), \quad (2)$$

where F_i is a projection operator [15] and is defined as

$$F_i = \frac{\mathbf{p}_i \mathbf{p}_i^t}{\mathbf{p}_i^t \mathbf{p}_i} = \frac{1}{\|\mathbf{p}_i\|^2} \begin{pmatrix} u_i^2 & u_i v_i & u_i \\ u_i v_i & v_i^2 & v_i \\ u_i & v_i & 1 \end{pmatrix}. \quad (3)$$

We refer to (1) as the *image space collinearity equation* and (2) as the *object space collinearity equation.*

The pose estimation problem is to find the rigid transform $(R, \mathbf{t})$ that best fits the known 3D object data with the observed 2D image data (see Figure 1). Usually this is achieved by minimizing some form of accumulation of errors (least squares methods) based on one of the collinearity equations in object or image space.

For transformation of a coordinate system A to another coordinate system B we represented the motion by a 4×4 homogeneous transformation matrix

$$T_A^B \cong \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}.$$

An alternative representation of the pose is by a six element vector $\mathbf{s} = (t_x, t_y, t_z, \theta, \phi, \psi)$ containing the three translational parameters and three angles of rotation around the three main axes. Equivalently quaternion representation can be used for orientation. In this paper f denotes the function which projects object points $\mathbf{P}_i$ to image points $\mathbf{p}_i = f(\mathbf{P}_i, \mathbf{s})$ in a camera with the pose specified by $\mathbf{s}$.

2.2 Previous Work

3D-2D registration in general is still a difficult unsolved problem in computer vision. Classical approaches make use of the ICP principle for pose or motion estimation [27, 3, 26]. The problem with the ICP based algorithms is that they converge to the closest local minimum, and thus not appropriate for solving large motion problems. We try to overcome this problem by coupling a global method to obtain a rough pose estimation using stationary cameras (see section 3.3)

The other way for the intialization is to obtain a set of initial registrations by sampling the 3-D orientation space, and then apply the algorithm to each initial registration. The optimal solution is the one with the global minimum error. This method was used by Besl and McKay [3] to solve the object recognition problem.

Formulating pose estimation as a nonlinear least squares problem, and solving it by nonlinear optimization algorithms is the classical approach used in photogrametry [Rosenfeld59, Tompson68, Haralick93]. Typically Gauss-Newton method or ... (Tamura) are used for this purpose. In computer vision Lowe used the Gauss-Newton method for the pose estimaton problem [Lowe87,92]. Starting from a good initial guess of the pose, the model is projected into the image plane and correspondences are found. This is done using a probabilistic approach to match image features with projected model entities. Having the correspondences, Gauss-Newton method is applied to determine the object rotation and translation. Most of such nonlinear optimization procedures rely on a good initial estimation of the pose.

Hager et al. propose a method to minimize a error metric in object space and then show that this function can be rewritten in a way which admits an iteration based on the solution to the 3D-3D pose estimation or *absolute orienation problem* [Haralick]. They formulated the pose estimation problem is as that of minimizing the collinearity error in object space rather than in the image space. We use a pose estimation method introduced by Hager et al. [15] to minimize the object space collinearity error (?). They propose an iterative algorithm which directly computes orthogonal rotation matrices, is fast and globally convergent.

The algorithm operates by successively improving an estimate of the rotation protion of the pose, and then estimates an assiciated translation. The intermediate rotation estimates are always the best "orthogonal" solution for each iteration. The orthogonality constraint is enforced by using singular value decomposition, and not from a specific parametrization of rotations e.g. euler angles. The iterative algorithm computes directly the orthogonal matrix and is proven to be globally convergent.

3. Overview of our Approach

The initialization of the tracking system is done as follows. First we estimate the position of the user's viewpoint with stationary cameras using a head tracking system and use the image taken by the mobile camera, referred to as the *initial* image, to estimate the initial orientation parameters (see section 3.3). The estimation is then refined by applying the optimization procedure described in section 3.4. During this estimation and refinement process particular attention is given to error propagation and statistical analysis.

To refine the pose data, we introduce a novel method for robust ICP based pose estimation based on defining of two collinearity constraints. They are used as a quality measurement for outliers detection and removal, therefore increasing the efficiency of the algorithm while providing the same accuracy as the classical method intrduced by Hager et al. [15].

Using a camera for tracking the greates uncertainty of the pose estimate is along the line of sight of the camera and the smallest error is perpendicular to this line. The reason is that a small translation of the camera parallel to image plane would result in an easily measurable change in the image where a small translation perpendicular to image plane generates only a small displacement in the image.

In our approach the data from mobile and stationary

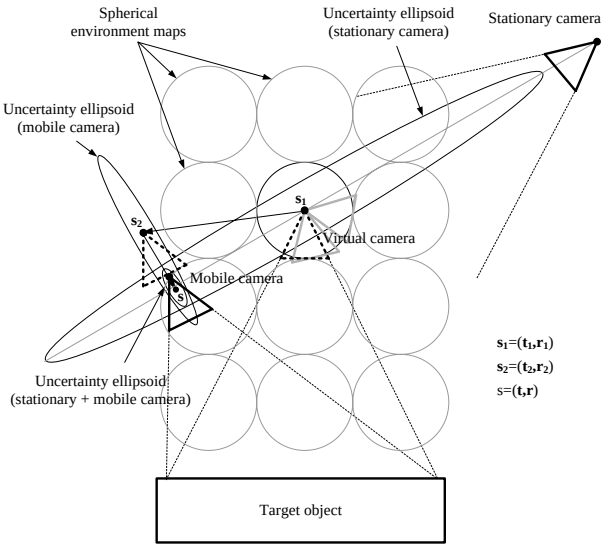

Figure 2. The iterative initial registration approach.

cameras are combined in order to estimate the pose of the user's view accurately.

Fig. 2 gives an overview of the automated initial registration approach with one stationary camera. The uncertainty ellipsoid determined by the stationary cameras shown in Figure 2 is a function of the characteristic of the external outside-in tracking system and the position of the user's viewpoint. Using more than one camera or even other tracking devices, only the shape of the uncertainty ellipsoid will change and requires no further handling in our system.

The main steps of our iterative initial registration algorithm are:

1. Estimating the initial position $\mathbf{t}_1 = (t_x, t_y, t_z)$ of the user's viewpoint using stationary cameras and its uncertainty as described in section 3.1 (Outside-In).

2. Estimating the initial orientation $\mathbf{r}_1 = (\theta, \phi, \psi)$ by extracting edges from the initial image and finding the best match of the edges in the initial image with the edges on the nearest environment map using a similarity measure (see section 3.3) (Inside-Out).

3. Establishing correspondences between 2D edges in the initial image and 3D edges on the 3D model and determining the relative pose $\mathbf{s}_2 = (\mathbf{t}_2, \mathbf{r}_2)$ and its associated error distribution (see section 3.4).

4. Statistical fusion of the two positional estimates $(\mathbf{s}_1, \mathbf{s}_2)$ from outside-in and inside-out cameras to $\mathbf{s}$ using their uncertainties (see section 3.2).

5. Go back to step 2 unless one of the following termination criteria is reached:

   (a) The displacement between image and model data is smaller than ϵ.

   (b) The change in motion parameters estimated in two consecutive iterations is smaller than $\Delta\epsilon$.

   (c) The maximal number of iterations is reached.

The thresholds ϵ and $\Delta\epsilon$ defined in the termination criteria depend on two requirements: First, the required precision of the initial pose for a successfull feature based tracking system which depends on the tracking solution used, and second, the time required for initialization which depends on the application the tracker is used for.

The system provides the uncertainty of the estimated pose in form of covariance matrix. The next section describes how the error estimation and propagation through different coordinate systems is done.

3.1 Error Estimation and Propagation

We assume that the noise in the image points is independent and its distribution is known by a covariance matrix $\Lambda_{\mathbf{pp}}$. Based on the edge detection algorithm used the covariance matrix $\Sigma_{\mathbf{pp}}$ could be estimated often based on the entries of the Hessian [22]. The uncertainty of the pose is represented by a 6×6 covariance matrix $\Lambda_{\mathbf{ss}}$. It is defined as $\Lambda_{\mathbf{ss}} = E(\Delta\mathbf{s}\Delta\mathbf{s}^t)$, the expectation of the square of the difference between the estimated $\tilde{\mathbf{s}}$ and the true values $\mathbf{s} = \tilde{\mathbf{s}} + \Delta\mathbf{s}$.

To compute the covariance matrix $\Lambda_{\mathbf{ss}}$ the nonlinear function f is linearized at the estimated pose $\tilde{\mathbf{s}}$ [10]. The Taylor series expansion gives after neglecting terms of the second and higher order:

$$\mathbf{p}_i + \Delta\mathbf{p}_i = f(\mathbf{P}_i, \tilde{\mathbf{s}} + \Delta\mathbf{s}) \approx f(\mathbf{P}_i, \tilde{\mathbf{s}}) + J_i\Delta\mathbf{s},$$

where $J_i = \left|\frac{\partial f}{\partial \mathbf{s}}\right|^t_{\mathbf{P}_i, \tilde{\mathbf{s}}}$ is the Jacobian of f evaluated at $(\mathbf{P}_i, \tilde{\mathbf{s}})$. Since $\mathbf{p}_i \approx f(\mathbf{P}_i, \tilde{\mathbf{s}})$, we get

$$\Delta\mathbf{p}_i = J_i\Delta\mathbf{x}.$$

Stacking all the equations for n points yields $\Delta P = J\Delta\mathbf{s}$. This equation can now be solved for $\Delta\mathbf{s}$ in a least squares manner as $\Delta\mathbf{s} = (J^tJ)^{-1}J^t\Delta P$. The covariance matrix is calculated by substituting $\Delta\mathbf{s}$:

$$\begin{aligned}\Lambda_{\mathbf{ss}} &= E(\Delta\mathbf{s}\Delta\mathbf{s}^t) \\ &= (J^tJ)^{-1}J^tE(\Delta P\Delta P^t)((J^tJ)^{-1}J^t)^t. \quad (4)\end{aligned}$$

Since we assume that the errors in image points are not correlated we have $E(\Delta\mathbf{p}_i\Delta\mathbf{p}_j) = 0$, for $i \neq j$. Therefore

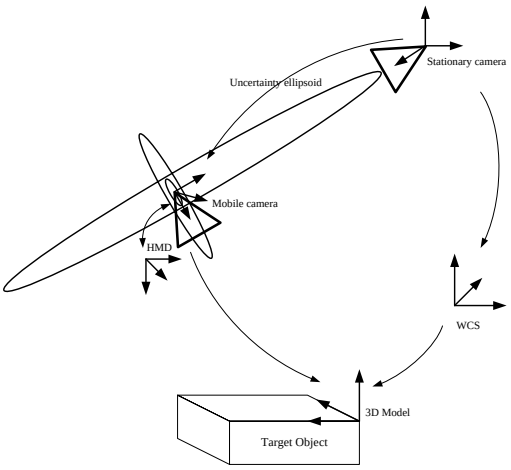

Figure 3. The coordinate systems.

$\Sigma_{\mathbf{PP}}$ is a diagonal matrix containing $\Sigma_{\mathbf{pp}}$ as diagonal elements. Using equation (4) the uncertainty of the pose can be estimated as a covariance matrix in the respective camera coordinate system.

Having different coordinate systems (see Fig. 3), we need to transform the covariance matrix properly to the same unit coordinate system. Propagating uncertainty in general through different functions is described by the *error propagation law* [12]. Given the pose $\mathbf{x}$ and its covariance matrix $\Lambda_{\mathbf{xx}}$, let $\mathbf{y} = g(\mathbf{x}) = T_A^B\mathbf{x}$ be the function which transforms $\mathbf{x}$ form coordinate system A to $\mathbf{y}$ in coordinate system B. The covariance matrix of $\mathbf{y}$ can then be calculated by

$$\Lambda_{\mathbf{yy}} = J\Lambda_{\mathbf{xx}}J^t, \text{ with } J = \frac{\partial g}{\partial \mathbf{x}}. \tag{5}$$

A useful representation of covariance matrices in 3D are the error ellipsoids, assuming that the errors are jointly gaussian. The joint probability density function (pdf) for N-dimensional error vector $\mathbf{x}$ is

$$p(\Delta\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N\,|\Lambda_{\mathbf{xx}}|}} e^{-\frac{1}{2}\Delta\mathbf{x}^t\Lambda_{\mathbf{xx}}^{-1}\Delta\mathbf{x}}.$$

If the argument of the exponent is constant, the surface of constant probability is an ellipsoid specified by the equation $\Delta\mathbf{x}^t\Lambda_{\mathbf{xx}}^{-1}\Delta\mathbf{x} = c^2$, for a constant c. For $c = 3$ the cumulative probability of the error vector $\mathbf{x}$ being inside the ellipsoid is approximately 97% [10].

3.2 Fusion of Pose Estimations

The pose parameters obtained using different tracking systems are fused in the following manner. Let $\mathbf{s_1}$ and $\mathbf{s_2}$ be the two pose vectors and $\Lambda_{\mathbf{s_1s_1}}$, $\Lambda_{\mathbf{s_2s_2}}$ the respective covariance matrices. The combined estimate $\mathbf{s}$ is obtained by weighting and averaging the covariance matrices [6, 10].

$$\mathbf{s} = (\Lambda_{\mathbf{s_1s_1}}^{-1} + \Lambda_{\mathbf{s_2s_2}}^{-1})^{-1}(\Lambda_{\mathbf{s_1s_1}}^{-1}\mathbf{s_1} + \Lambda_{\mathbf{s_2s_2}}^{-1}\mathbf{s_2}) \tag{6}$$

Reforming this equation yields for $\mathbf{s}$ and its covariance matrix

$$\begin{aligned} \mathbf{s} &= \Lambda_{\mathbf{s_2s_2}}(\Lambda_{\mathbf{s_1s_1}} + \Lambda_{\mathbf{s_2s_2}})^{-1}\mathbf{s_1} + \\ &\quad \Lambda_{\mathbf{s_1s_1}}(\Lambda_{\mathbf{s_1s_1}} + \Lambda_{\mathbf{s_2s_2}})^{-1}\mathbf{s_2}, \\ \Lambda_{\mathbf{ss}} &= \Lambda_{\mathbf{s_2s_2}}(\Lambda_{\mathbf{s_1s_1}} + \Lambda_{\mathbf{s_2s_2}})^{-1}\Lambda_{\mathbf{s_1s_1}}. \end{aligned} \tag{7}$$

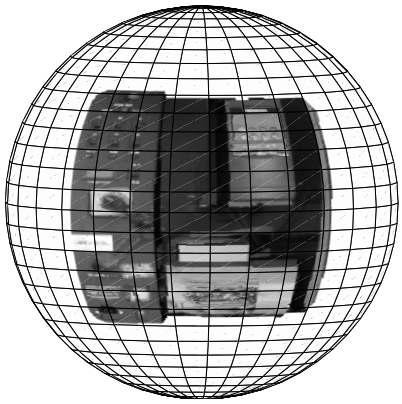

Figure 4. The spherical environment map.

3.3 Coarse Registration Using Environment Maps

In computer vision Adelson and Bergen [1, 16] assigned the name *plenoptic function* (from plenus, complete or full, and optic) to the pencil of rays visible from any point in space, at any time, and over any range of wavelengths. They use this function to develop a taxonomy for evaluating models of low-level vision. The plenoptic function is a parameterized function for describing everything that can be seen from the point of view of the user. From a given point of view we can select any of the viewable rays by choosing an azimuth and elevation angle (θ, ϕ). In computer graphics terminology, the plenoptic function describes the set of all possible environment maps for a given scene.

We first define a complete sample of the plenoptic function as a full spherical map for a given viewpoint, defined thanks to the external outside-in tracking cameras. Having a virtual model of the environment or the target objects, the viewing space can be coarsely sampled and a set of spherical environment maps $\mathbf{M} = \{\mathbf{M}_i | 0 < i \leq N\}$ is generated. [1]

[1] Due to the comlexity of the scene and the samle rate this can be a quite time consuming procedure. This does not affect the computational cost of the system since this can be done offline using a redering system.

For the first stage of the initialization procedure the orientation of user's head is estimated as following. Given a set of samples $\mathbf{M}$ from the plenoptic function in form of spherical environment maps, the system selects the closest environment map $\mathbf{M}_k \in \mathbf{M}$. The initial view is then projected onto the spherical map $\mathbf{M}_k$, and the best match represented by three rotational parameters is estimated using a similarity measure.

Several methods has been proposed in the literature to align two 2D images [review registration]. We use a intensity based similarity measure known as the gradient correlation [?]. For this purpose four gradient images are created by horizontal and vertical Sobel templates from the respective environment map and the initial image. The normalized cross correlation (NCC) is calculated of both horizontal and vertical gradient images, respectively. The similarity measure value is the average of the two NCCs. We use an efficient implementation for fast computation of NCC [14, 9].

Since the gradient images are used for registration we only need to save the gradient images of the environments maps. To reduce the amount of data storage needed for storing the gradient images Laplacian pyramid [?] can be used. In order to speed up the searching for the highest correlation value, the same image pyramids can be used. This results in a hierarchical iterative registration of the initial image and the respective spherical environment map.

Due to the uncertainty in estimating user's viewpoint from stationary cameras and the limited sampling rate of the plenoptic function only a coarse pose estimation can be achieved with the method described above. We therefor use this estimated pose as the initial values for a second stage of the initialization, in which a 3D-2D pose estimation method is proposed to accurately estimate the relative displacement of the pose.

3.4 Refined 2D-3D Registration

The coarse pose estimation brings the contour edges extracted in the initial frame close to the corresponding edges in the respective environment map. The extracted contour edges are represented as a set of discrete points. Using the 3D model of the scene, we can retrieve the 3D position of the edge points on the virtual model. This section describes how to determine the relative pose that coincides the position of those 3D edge points on the model onto the 2D edges points in the initial image plane, i.e. registration of 3D and 2D points. For more robustness small edges are removed by thresholding and only dominant edges are used.

This part of the initialization procedure has to be not only accurate, but also robust and computationally efficient.

Since we have no a-priori knowledge of correspondences we use a 3D-2D registration algorithm based on the iterative closest point (ICP) principle [27, 3]. It is composed of three iterated steps, the first of which determines correspondence candidates between 2D and 3D edge points. In the second step a robust technique is used to discard the outliers by analyzing the statistics of the distances. And finally the third step estimates the 3D rigid transformation that minimizes the displacement of matched points.

The next three sections describe each step of the algorithm.

3.4.1 Establishing Correspondence Candidates

Since there is no distance metric relating the 2D edge points in the initial frame to 3D edge points on the model, there is no obvious way to directly applying the ICP principle, to the registration of 3D model edge points to 2D image edge points.

Let $\{\mathbf{p}'_j | 0 < j \leq m\}$ denote the set of extracted 2D image edge points. The correspondence candidates are chosen in a way that both the 2D error distance between the back projected model edge point and observed image points $\mathbf{p}'_j$ as well as the 3D distance of the model points to the projection rays of $\mathbf{p}'_j$ in object space is minimized.

Given the estimated pose parameters $\tilde{R}$ and $\tilde{\mathbf{t}}$, the distance between a 3D model point $\tilde{R}\mathbf{P}_i + \tilde{\mathbf{t}}$ to the projection ray of the 2D edge point $\mathbf{p}'_j$ is due to object space collinearity equation (2)

$$d_{obj}(\mathbf{P}_i, \mathbf{p}'_j) = \|\mathbf{Q}_i - F_j\mathbf{Q}_i\| = \left\|(I - F'_j)(\tilde{R}\mathbf{P}_i + \tilde{\mathbf{t}})\right\|, \tag{8}$$

where F'_j is the projection operator (see section ?) defined for the image points $\mathbf{p}'_j$

$$F'_j = \frac{\mathbf{p}'_j\mathbf{p}'^t_j}{\mathbf{p}'^t_j\mathbf{p}'_j}.$$

Analogue due to the image space collinearity equation (1) the distance between the 2D edge point and the back projected 3D model point on the image plane is

$$d_{img}(\mathbf{P}_i, \mathbf{p}'_j) = \left\|\mathbf{p}'_j - \frac{\tilde{R}\mathbf{P}_i + \tilde{\mathbf{t}}}{\tilde{\mathbf{r}}_3\mathbf{P}_i + \tilde{t_z}}\right\|. \tag{9}$$

The smaller d_{obj} and d_{img}, the more likely $(\mathbf{P}_i, \mathbf{p}'_j)$ represent a correct correspondence. If $(\mathbf{P}_i, \mathbf{p}'_j)$ and $(\mathbf{P}'_{c_j}, \mathbf{p}'_j)$ have the same object space error $d_{obj} = d'_{obj}$ then $(\mathbf{P}'_i, \mathbf{p}_i)$ should be preferred because $d_{img} > d'_{img}$. See Figure ?. To establish correspondence candidates for every image point we take the model edge points with the smallest distance d defined as the sum of the both distances in image and object space.

$$d(\mathbf{P}_i, \mathbf{p'}_j) = \alpha_{img}d_{img}(\mathbf{P}_i, \mathbf{p'}_j) + \alpha_{obj}d_{obj}(\mathbf{P}_i, \mathbf{p'}_j).$$

Since the distances in image and object space are of different order they are weighted properly using the factors α_{img}

and α_{obj}. For $\alpha_{img} = 1$ and $\alpha_{obj} = 0$ only the nearest points in the image are considered as correspodence used by [zhang] whereas for $\alpha_{img} = 0$ and $\alpha_{obj} = 1$ only the nearest point in the object space are chosen as candidates [Hager]. We use both distances to select the best matches for this purpose.

I.e. for any 2D edge point $\mathbf{p}'_j$ $(0 < j \leq m)$, its correspondence candidate $\mathbf{P}_{c_j}$ is determined as

$$\mathbf{P}_{c_j} = argmin_{c_j \in \{1,\ldots,n\}}\, d(\mathbf{P}_{c_j}, \mathbf{p'}_j), \qquad (10)$$

where c_j are the corresponding indices of the 3D model points. The search space is determined by the size of the model edge point set n. In order to speed up the search, optimized K-D tree data structure can be used to accelerate the closest point search [19, 27].

3.4.2 Estimating Motion

This section describes briefly how to re-estimate the pose parameters that minimize the displacement between the corresponding edge points.

According to the ICP principle we minimize the object space collinearity error (2) by moving the model data $\mathbf{P}_{c_j}$ such that at each step the displacement between $\mathbf{p}_j$ and $\mathbf{P}_{c_j}$ is minimized.

The basic idea is to reduce at each iteration the 3D-2D registration problem to the 3D-3D registration of points by using the 3D projection points on the respective image rays instead of 2D image points **??**. For each iteration the projection points have to be determined again due the new pose estimates.

Formally, we seek the rigid pose parameters R and $\mathbf{t}$ that minimizes the following mean-squares objective function

$$E(R, \mathbf{t}) = \sum_{j=1}^{m} w_j \left\| (I - F'_j)(R\mathbf{P}_{c_j} + \mathbf{t}) \right\|^2, \qquad (11)$$

subject to the orthogonality constraint $RR^t = I$. The w_j are positive weighting factors associated with each correspondence candidate. See next section for how to choose these weights.

For a fixed rotation R the optimal translation t can be computed from (11) as

$$\mathbf{t}(R) = (I - \frac{1}{m}\sum_{j=1}^{m} F'_j)^{-1} \sum_{j=1}^{m} (F'_j - I) R\mathbf{P}_{c_j} \qquad (12)$$

The estimated rotation matrix $\tilde{R}$ can be used as the starting point and is re-estimated iteratively as follows. Let R^k be the kth estimate of R, $\mathbf{t}^{(k)} = \mathbf{t}(R^{(k)})$, and $\mathbf{Q}^{(k)}_{c_j} = R^{(k)}\mathbf{P}_{c_j} + \mathbf{t}^{(k)}$. The next estimate $R^{(k+1)}$ is determined by

$$R^{k+1} = argmin_R \sum_{j=1}^{m} w_j \left\| R\mathbf{P}_{c_j} + t^{(k)} - F'_j \mathbf{Q}^{(k)}_{c_j} \right\|^2, \qquad (13)$$

subject to $R^t R = I$. Such a constrained least squares problem can be solved for $R^{(k+1)}$ in closed form using quaternions [11] or singular value decomposition (SVD) [8]. For the SVD solution, first a sample cross-covariance matrix M between $\mathbf{P}_{c_j}$ and $\mathbf{L}^{(k)}_{c_j} = F'_j \mathbf{Q}^{(k)}_{c_j}$ is calculated.

$$M = \sum_{j=1}^{m} w_j (\mathbf{P}_{c_j} - \bar{\mathbf{P}})(\mathbf{L}^{(k)}_{c_j} - \bar{\mathbf{L}}^{(k)}), \qquad (14)$$

where $\bar{\mathbf{P}}$ and $\bar{\mathbf{L}}^{(k)}$ are the centroids, respectively. Let UDV^t be a SVD of M, where U and V are orthogonal matrices, and D is diagonal. Then the optimal solution to (13) is

$$R^{(k+1)} = VU^t. \qquad (15)$$

The next estimate of translation is then computed by $\mathbf{t}^{(k+1)} = \mathbf{t}(R^{(k+1)})$ from (12). Note, that the computational complexity at each iteration k is linear in the number of points considered. This iterative algorithm directly computes orthogonal rotation matrices, it is fast and convergent due to the convergence theorem of the original ICP algorithm [3].

3.4.3 Robust Pose Estimation

Due to both occlusion and inaccuracy in pose estimation, correspondence candidates $(\mathbf{p}'_j, \mathbf{P}_{c_j})$ established in the section 3.4.1 are not guaranteed to be correct. There are two types of outliers (wrong matches). The first is where $\mathbf{P}_{c_j}$ is not the corresponding 3D model point, while the correct correspondence exist. The second is a match where the corresondencing point is not included in the set of 3D model points.

A large number of algorithms described above have been developed to quantitatively evaluate the 2D-2D or 3D-3D point matches. In this section we describe a novel approach to evaluate the 3D-2D point matches based on a statistical model.

The equation (2) and (1) essentially describe an object space and an image space collinearity constraint. The former means that the image point $\mathbf{p}'_j$, the projection of $\tilde{R}\mathbf{P}_{c_j} + \tilde{\mathbf{t}}$ on $\mathbf{p}'_j$ and the optical center $\mathbf{O}$ of the camera are in as collinear as possible. The latter constraint means that the the image point $\mathbf{p}'_j$, model point $\mathbf{P}_{c_j}$ and projection center are collinear.

These two constraint imply that collinearity error both in object space d_{obj} (8) and in image space d_{img} (9) should be minimized.

These constraints represent necessary conditions for a pair of 2D-3D edge points to be correct. If a candidate does not satisfy any of these constraints, it can not be a correct one. Thus, we use these constraints as a quality measurement of correspondence candidates from which relatively good matches can be selected and used for pose re-estimation.

Based on the point matches the means $\mu_{d_{obj}}$ and $\mu_{d_{img}}$ and standard deviations $\sigma_{d_{obj}}$ and $\sigma_{d_{img}}$ are computed. Depending on these values we reject outliers:

$$\begin{aligned} \text{if } (\left|d_{obj}(\mathbf{p}'_j, \mathbf{P}_{c_j}) - \mu_{d_{obj}}\right| > \kappa_{obj}\sigma_{d_{obj}} \text{ or} \\ \left|d_{img}(\mathbf{p}'_j, \mathbf{P}_{c_j}) - \mu_{d_{img}}\right| > \kappa_{img}\sigma_{d_{img}}) \\ \text{then } w_j = 0. \end{aligned} \tag{16}$$

where w_j is a weighting factor introduced in equation (11). The maximum tolerance parameters κ_{obj} and κ_{img} represent how many per cent of matches are cosidered as outliers and rejected. For a parameter value of 1.0 appriximately 68% of matches lying in the interval $[\mu - \sigma, \mu + \sigma]$ are taken into account and the rest is rejected as outliers. For a value of 3 all the matches are used for motion estimation. The maximum tolerable distances in image and object space can be detemined based on the maximal motion expected. They can be chosen properly based on the scene depth and sampling rate of the plenoptic function. This values have an impact on the convergence of the algorithm. If they are too small, more iterations are required for the algorithm because many good correspondence candidates will be discarded.

As a result of this procedure, a set of refined correspondences will be obtained.

Another problem that arises with the solution described in the previous section is that if the image points are perturbed by homogeneous gaussian noise, the pose solution will implicitly more heavily weight model points that are farther away from the camera, since the d_{obj} increases with distance of the model point to the camera. Supposing that the residual error, i.e. the distance d_{obj} is approximately proportional to the depth and equal for all points, the weights w_j can be chosen as

$$w_j = \frac{1}{(Z_{c_j}^{(k)})^2}, \tag{17}$$

where $Z_{c_j}^{(k)}$ is the depth of each model point $Q_{c_j}^{(k)}$ in the camera coordinate system [15].

4. Experimental Results

The proposed registration algorithm has been tested in a number of both simulated and real scenes and the registration accuracy was analyzed. Because of lack of space we present the results of the real experiments in this paper only.

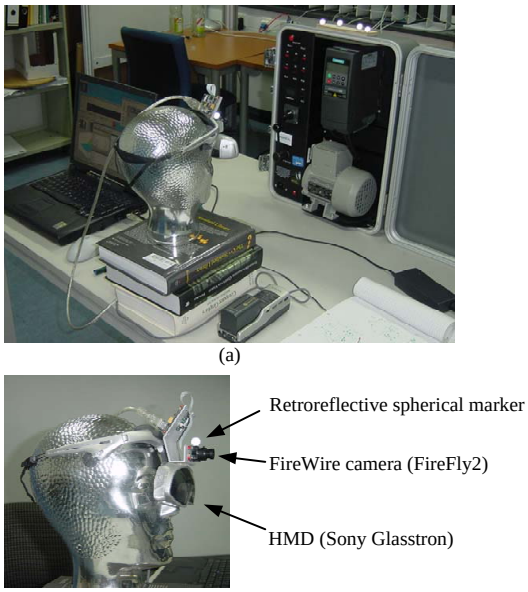

Figure 5. The Experimental Setup.

The registration algorithm has been implemented in Matlab on a 2GHz Pentium 4 CPU, with 1.0 GB RAM. The code is not optimized yet because we are in testing and evaluation phase. The algorithms are however designed with particular attention to real-time requirement.

In our experimental setup (see Figure 5(a)) a FireWire (IEEE-1394) digital camera (FireFly2 from Point Grey) was mounted on the optical see-through HMD (Sony Glasstron). We used a 4 mm wide angle lens with a field-of-view of 68 degrees for the experiments. The camera was internally calibrated and lens distortion was corrected using the camera calibration toolbox [4].

For evaluation purposes we use the A.R.T. [2] outside-in tracking system in our AR lab to track the position of the user's viewpoint. Thereby, a small retroreflective spherical marker is attached to the mobile camera (see Fig. 5(b)), which was then tracked by three ART cameras hanging in the corners of our lab. The tracking data were sent via wireless LAN to the mobile computer with the application running. In the future this marker-based system will be replaced by a marker-less head tracking system using stationary smart cameras.

As an AR scenario, we used a control unit box provided by Siemens Automation as the target object. We created an accurate 3D model of the box using the software ImageModeler from RealViz [20] (see Fig. 6(a)). ImageModeler uses photographic images taken from the object to recover the 3D geometry and maps automatically the original images onto the model's surface as texture maps, resulting in a highly realistic model.

The pose of the box in the room was determined by plac-

(a) 3D model

(b) Cylinderical environment map

(c) Best match of the initial image on the environment map

(d) Initial image

(e) Virtual camera view

Figure 6. The coarse registration.

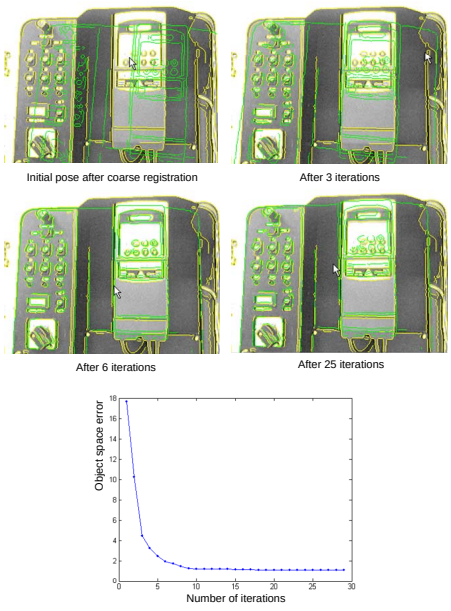

Figure 7. Aligned model edges with the edges of the initial image.

ing a ART-target at a fix position on the box. The transformation between the ART marker coordinate frame and the model frame was calculated by measuring the 3D coordinates of some points on the box in model and marker coordinate frame respectively. From those correspondences the transformation parameters were calculated. This step needs to be done only once when the box is moved relative to the ART tracking cameras.

For projecting a complete plenoptic sample the most natural surface would be a unit sphere centered about the viewing position (see section 3.3). However, the difficulty of spherical projections is the lack of a representation that is suitable for data storage, particularly for a uniform discrete sampling [16]. We have therefore chosen to use a cylindrical projection as the plenoptic sample representation. The advantage of a cylinder is that it can be easily unrolled into a simple planar map.

Figure 6(b) shows a cylindrical environment map of the box as a sequence of images taken from panning the virtual camera 360 degrees around the optical axis. Thereby the internal parameters of the virtual camera are identical to the real camera used.

The viewing space in front of the virtual box was sampled automatically every 5 cm in each space direction, resulting in a total set of 1080 environment maps, where only the respective gradient images were stored.

Figure 6(d) shows the initial image of size 320×240 taken by the mobile camera after the lens distortion correction.

Using the A.R.T. outside-in tracker the position of the marker attached to the camera was determined and the nearest environment map was selected automatically. At a range of 1.5 m, the stated RMS (root mean square) accuracy by the manufacturer of locating a single ART marker is 0.5 mm in all directions. Since the marker is about 3.5 cm away from the camera, we assume the positional error is about 4 cm.

In our experiments with three tracking cameras, the positional uncertainty could be modeled with an error ellipsoid of the form of a sphere since the variaitions are not statistically significant in this study.

A coarse estimation of the orientation was then done by finding the best match of the initial image in the respective environment map (see Figure 6(c)). From the position of the best match the azimuth and elevation angle were derived. In our experiments elevation and the rotation angle (around the optical axis of the camera) are considered to be small. Up to a maximum variation of ± 30 degrees the correct match could be found properly using the NCC measurement (see section 3.3).

The view of the virtual camera with the coarse orientation is shown in Fig. 6(e). The extracted edges of the virtual camera image are superimposed in Fig. 7(a) on the initial image. The final displacement is then estimated using the

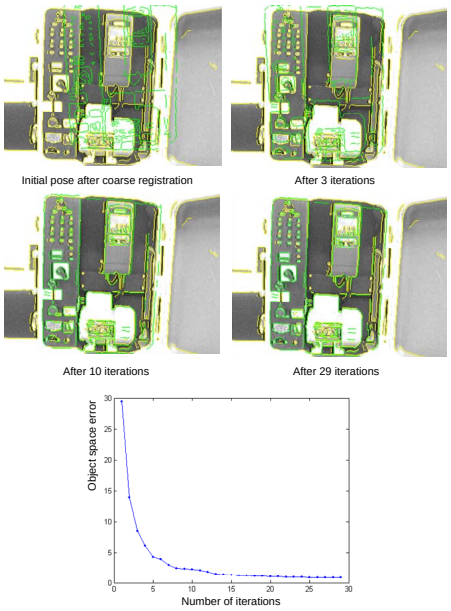

Figure 8. Aligned model edges with the edges of the initial image.

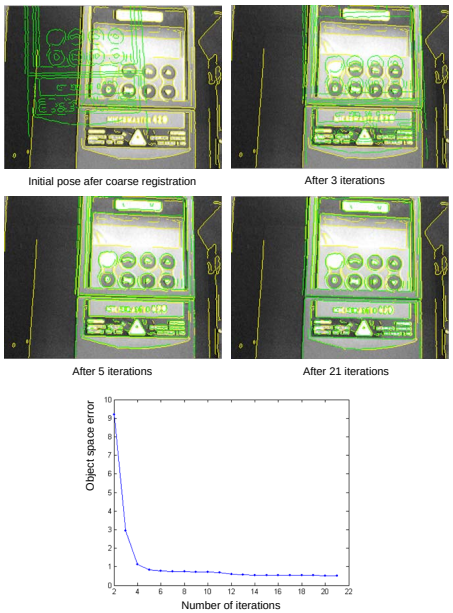

Figure 9. Aligned model edges with the edges of the initial image.

method described in section 3.4. For this purpose the motion between the 2D edges in the initial image and 3D edges on the model is estimated.

Figure 9(a)-(d) shows the procedure of the iterative registration of the 3D edges with 2D edges. Figure 9(e) shows a plot of the the object space error d_{obj} defined in (8), during the pose estimation process. We observe a fast convergence of the algorithm during the first iterations that slows down as it approaches its minimum.

The uncertainty of the final pose derived from the mobile camera was then estimated as a covariance matrix as described in section 3.1. The maximum translational error along the line of sight is about 21 mm. The error ellipsoid of the overall estimation after fusion has a major axis of 1.6 mm.

Figure 8 and 9 show the registration process of two different initial views of the target object.

5. Conclusions and Future Work

We presented a sensor fusion approach for automated initilization of marker-less tracking systems. This was achieved by analyzing and estimating the error of tracking sensors. The uncertainty of the tracking sensors is represented by covariance matrices and can be visualized as 3D ellipsoids. The initial pose was then estimated iteratively with a coarse to fine strategy by taking the uncertainties into account. We applied the method to a an augmented reality system using mobile and stationary cameras.

The pose parameters are estimated in two estimation and refinement steps. Thereby the second step is independent of the first one and can converge in worst case to a wrong local minimum. This can be prevented in a future work by considering the positional uncertainty drived from the stationary cameras during the refined pose estimation step.

The pose refinement in the second step is based on minimizing the object space collinearity errors. A more effiecient and maybe faster solution could be the minimization of both object and image space collinearity errors simultanously.

Acknowledgements. Hesam Najafi is supported by a scholarship funded from Siemens Corporate Research. We also thank Siemens Automation & Drive for providing the control unit box.

References

[1] E. H. Adelson and J. R. Bergen. *Computational Models of Visual Processing, Chapter 1: The Plenoptic Function and the Elements of Early Vision*. The MIT Press, Cambridge, Mass, 1991.

[2] ART. Advaned real time tracking. *www.ar-tracking.com*.

[3] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–255, 1992.

[4] J.-Y. Bouguet. Camera calibration toolbox for matlab. *http://www.vision.caltech.edu/bouguetj/calib*

doc/index.html, The C implementation of this toolbox is included in the Open Source Computer Vision library distributed by Intel.

[5] K. Chia, A. Cheok, and S. Prince. Online 6dof augmented reality registration from natural features. *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, October 2002.

[6] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, 1993.

[7] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar, and N. Navab. Marker-less tracking for ar: A learning-based approach. *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, October.

[8] R. M. Haralick, H. Joo, C.-N. Lee, X. Zhuang, and M. B. Kim. Pose estimation from corresponding point data. *IEEE Trans. Systems, Man, and Cybernetics*, 6(19), November 1989.

[9] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision, Volume II*. Addison-Wesley, 1992.

[10] W. Hoff and T. Vincent. Analysis of head pose accuracy in augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 6(4), October-December 2000.

[11] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternion. *J. Opt. Soc. Amer.*, A-4:629–642, 1987.

[12] K.-R. Koch. *Parameter Estimation and Hypothesis Testing in Linear Models*. Springer Verlag, Berlin, Heidelberg, 1999.

[13] V. Lepetit, L. Vacchetti, , and P. Fua. Fully automated and stable registration for augmented reality applications. *International Symposium on Mixed and Augmented Reality (ISMAR'03)*, September 2003.

[14] J. P. Lewis. Fast normalized cross-correlation. *Industrial Light & Magic*.

[15] C.-P. Lu, G. D. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE PAMI*, 22(6):610–622, 2000.

[16] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Proceedings of SIGGRAPH'95, Computer Graphics Proceedings, Annual Conference Series*, pages 39–46, August 1995.

[17] H. Najafi and G. Klinker. Model-based tracking with stereovision for ar. *International Symposium on Mixed and Augmented Reality (ISMAR'03)*, October 2003.

[18] U. Neumann and S. You. Natural feature tracking for augmented-reality. *IEEE Transactions on Multimedia*, 1(1), 1999.

[19] F. Preparata and M. Shamos. *Computational Geometry, An Introcusion*. New York: Springer, Berlin, Heidelberg, 1986.

[20] RealViz. *www.realviz.com*.

[21] K. Satoh, S. Uchiyama, H. Yamamoto, and H. Tamura. Robust vision-based registration utilizing bird's-eye view with user's view. *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'03)*, October 2003.

[22] J. Shi and C. Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994.

[23] G. Simon, A. Fitzgibbon, and Z. A. Markerless tracking using planar structures in the scene. *International Symposium on Augmented Reality (ISMAR'00)*, October 2000.

[24] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[25] M. Wagner, A. MacWilliams, M. Bauer, G. Klinker, J. Newman, T. Pintaric, and D. Schmalstieg. Fundamentals of ubiquitous tracking. *Second International Conference on Pervasive Computing, Hot Spots section*, 2004.

[26] P. Wunsch and G. Hirzinger. Registration of cad-models to images by iterative inverse perspective matching. *13th International Conference on Pattern Recognition*, pages 77–83, August 1996.

[27] Z. Zhang. Iterative point matching for registration of free-form curves. *Technical Report, Inria-Sophia Antipolis*, (1658), March 1992.