Analysis of Head Pose Accuracy in Augmented Reality

William Hoff, *Member*, *IEEE*, and Tyrone Vincent, *Member*, *IEEE*

Abstract—A method is developed to analyze the accuracy of the relative head-to-object position and orientation (pose) in augmented reality systems with head-mounted displays. From probabilistic estimates of the errors in optical tracking sensors, the uncertainty in head-to-object pose can be computed in the form of a covariance matrix. The positional uncertainty can be visualized as a 3D ellipsoid. One useful benefit of having an explicit representation of uncertainty is that we can fuse sensor data from a combination of fixed and head-mounted sensors in order to improve the overall registration accuracy. The method was applied to the analysis of an experimental augmented reality system, incorporating an optical see-through head-mounted display, a head-mounted CCD camera, and a fixed optical tracking sensor. The uncertainty of the pose of a movable object with respect to the head-mounted display was analyzed. By using both fixed and head mounted sensors, we produced a pose estimate that is significantly more accurate than that produced by either sensor acting alone.

Index Terms—Augmented reality, pose estimation, registration, uncertainty analysis, error propagation, calibration.

1 Introduction

Augmented reality is a term used to describe systems in which computer-generated information is superimposed on top of the real world [1]. One form of enhancement is to use computer-generated graphics to add virtual objects (such as labels or wire-frame models) to the existing real world scene. Typically, the user views the graphics with a head-mounted display (HMD), although some systems have been developed that use a fixed monitor (e.g., [2], [3], [4], [5]). The combining of computer-generated graphics with real-world images may be accomplished with either optical [6], [7], [8] or video technologies [9], [10].

A basic requirement for an AR system is to accurately align virtual and real-world objects so that they appear to coexist in the same space and merge together seamlessly. This requires that the system accurately sense the position and orientation (pose) of the real world object with respect to the user's head. If the estimated pose of the object is inaccurate, the real and virtual objects may not be registered correctly. For example, a virtual wire-frame model could appear to float some distance away from the real object. This is clearly unacceptable in applications where the user is trying to understand the relationship between real and virtual objects. Registration inaccuracy is one of the most important problems limiting augmented reality applications today [11].

This paper shows how one can estimate the registration accuracy in an augmented reality system, based on the characteristics of the sensors used in the system. Only quasi-static registration is considered in this paper; that is, objects are stationary when viewed, but can freely be moved. We develop an analytical model and show how the model can be used to properly combine data from multiple sensors to improve registration accuracy and gain insight into the effects of object and sensor geometry and configuration. A preliminary version of this paper was presented at the First International Workshop on Augmented Reality [12].

• *The authors are with the Engineering Division, Colorado School of Mines, 1500 Illinois St., Golden, CO 80401.*
E-mail: {whoff, tvincent}@mines.edu.

Manuscript received 1 Feb. 1999; revised 6 July 2000; accepted 10 July 2000. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number 109094.

1.1 Registration Techniques in Augmented Reality

To determine the pose of an object with respect to the user's head, tracking sensors are necessary. Sensor technologies that have been used in the past include mechanical, magnetic, acoustic, and optical [13]. We concentrate on optical sensors (such as cameras and photo-effect sensors) since they have the best overall combination of speed, accuracy, and range [7], [14], [15].

There has been much work in the past in the photogrammetry and computer vision fields on methods for object recognition and pose estimation from images. Some difficult problems (which are not addressed here) include how to extract features from the images and determine the correspondence between extracted image features and features on the object. In many practical applications, these problems can be alleviated by preplacing distinctive optical targets, such as light emitting diodes (LEDs) or passive fiducial markings, in known positions on the object. The 3D locations of the target points on the object must be carefully measured, in some coordinate frame attached to the object. In this paper, we will assume that point features have been extracted and the correspondences known so that the only remaining problem is to determine the pose of the object with respect to the HMD.

One issue is whether the measured points are two-dimensional (2D) or three-dimensional (3D). Simple passive optical sensors, such as video cameras and photo-effect sensors, can only sense the direction to a target point and not its range. The measured data points are 2D, i.e., they

represent the locations of the target points projected onto the image plane. On the other hand, active sensors, such as laser range finders, can directly measure direction and range, yielding fully 3D target points. Another way to obtain 3D data is to use triangulation; for example, by using two or more passive sensors (stereo vision). The accuracy of locating the point is improved by increasing the separation (baseline) between the sensors.

Once the locations of the target points have been determined (either 2D or 3D), the next step is to determine the full six degree-of-freedom (DOF) pose of the object with respect to the sensor. Again, we assume that we know the correspondence of the measured points to the known 3D points on the object model. If one has 3D point data, this procedure is known as the "absolute orientation" problem in the photogrammetry literature. If one has 2D target points, this procedure is known as the "exterior orientation" problem [16].

Another issue is where to locate the sensor and target. One possibility is to mount the sensor at a fixed known location in the environment and put targets on both the HMD and on the object of interest (a configuration called "outside-in" [14]). We measure the pose of the HMD with respect to the sensor, and the pose of the object with respect to the sensor, and derive the relative pose of the object with respect to the HMD. Another possibility is to mount the sensor on the HMD and the target on the object of interest (a configuration called "inside-out"). We measure the pose of the object with respect to the sensor and use the known sensor-to-HMD pose to derive the relative pose of the object with respect to the HMD. Both approaches have been tried in the past and each has advantages and disadvantages.

With a fixed sensor (outside-in approach), there is no limitation on size and weight of the sensor. Multiple cameras can be used, with a large baseline, to achieve highly accurate 3D measurements via triangulation. For example, commercial optical measurement systems, such as Northern Digital's Optotrak, have baselines of approximately 1 meter and are able to measure the 3D positions of LED markers to an accuracy of approximately 0.15 mm. The orientation and position of a target pattern is then derived from the individual point positions. A disadvantage with this approach is that head orientation must be inferred indirectly from the point positions.

The inside-out approach has good registration accuracy because a slight rotation of a head-mounted camera causes a large shift of a fixed target in the image. However, a disadvantage of this approach is that large translation errors occur along the line of sight of the camera. To avoid this, additional cameras could be added with lines of sight orthogonal to each other.

1.2 Need for Accuracy Analysis and Fusion

In order to design an augmented reality system that meets the registration requirements for a given application, we would like to be able to estimate the registration accuracy for a given sensor configuration. For example, we would like to estimate the probability distribution of the 3D error distance between a generated virtual point and a corresponding real object point. Another measure of interest is the overlay error; that is, the 2D distance between the projected virtual point and the projected real point on the HMD image plane, which is similar to the image alignment error metrics that appear in other work [7], [9], [17].

Another reason to have an analytical representation of uncertainty is for fusing data from multiple sensors. For example, data from head-mounted and fixed sensors might be combined to derive a more accurate estimate of object-to-HMD pose. The uncertainties of these two sensors might be complementary so that, by combining them, we can derive a pose that is much more accurate than that from each sensor used alone. In order to do this, a mathematical analysis is required of uncertainties associated with the measurements and derived poses. Effectively, we can create a hybrid system that combines the "inside-out" and "outside-in" approaches.

1.3 Relationship to Past Work and Specific Contributions

Augmented reality is a relatively new field, but the problem of registration has received ample attention, with a number of authors taking an optical approach. Some researchers have used photocells or photo-effect sensors which track light-emitting diodes (LEDs) placed on the head, object of interest, or both [7], [14], [15]. Other researchers have used cameras and computer vision techniques to detect LEDs or passive fiducial markings [5], [8], [18], [19], [20], [21]. The resulting detected features, however they are obtained, are used to determine the relative pose of the object to the HMD. A number of researchers have evaluated their registration accuracy experimentally [17], [7], with Monte-Carlo simulations [19], or both [18]. However, no one has studied the effect of sensor-to-target configuration on registration accuracy. In this paper, we develop an analytical model to show how sensor errors propagate through to registration errors, given a statistical distribution of the sensor errors and the sensor-to-target configuration.

Some researchers avoid the problem of determining pose altogether and instead concentrate on aligning the 2D image points using affine projections [22], [23]. Although this approach works well for video-based augmented reality systems, in optical see-through HMD systems, it would not work as well because the image as seen by the head-mounted camera may be different than the image seen by the user directly through the optical combiner.

A number of researchers have developed error models for HMD-based augmented reality systems. Some researchers have looked at the optical characteristics of HMDs in order to calculate viewing transformations and calibration techniques [24], [25]. Holloway [17] analyzed the causes of registration error in a see-through HMD system, due to the effects of misalignment, delay, and tracker error. However, he did not analyze the causes of tracker error, merely its effect on the overall registration accuracy. This work, on the other hand, focuses specifically on the tracker error and does not look at the errors in other parts of the system, or attempt to derive an overall end-to-end error model.

In the computer vision field, the problem of determining the position and orientation from a set of given point or line correspondences has been well-studied. Some researchers have developed analytical expressions for the uncertainty of a 3D feature position as derived from image data [26]. Other

researchers have evaluated the accuracy of pose estimation algorithms using Monte Carlo simulations [27], [28], [29], [30]. Few researchers have addressed the issue of error propagation in pose estimation. We follow the method suggested by Haralick and Shapiro [16], who outline how to derive the uncertainty of an estimated quantity (such as a pose) from the given uncertainties in the measured data.

Kalman filtering [31] is a standard technique for optimal estimation. It has been used to estimate head pose in augmented and virtual reality applications [7], [32], [33]. From a sequence of sensor measurements, these techniques also estimate the uncertainty of the head pose. This is similar to the work described in this paper in the sense that a Kalman filter can be interpreted as a method for obtaining a maximum likelihood estimate of the state in a dynamic system, given input-output data [34]. Our system is static and so we do not have a model of the state dynamics. We fuse data from two measurements, rather than data from a measurement and a prediction from past data.

In this work, a method is developed to explicitly compute uncertainties of pose estimates, propagate these uncertainties from one coordinate system to another, and fuse pose estimates from multiple sensors. The contribution of this work is the application of this method to the registration problem in augmented reality. Specifically:

- The method shows how to estimate the uncertainty of object-to-HMD pose from the geometric configuration of the optical sensors and the pose estimation algorithms used. To help illustrate the method, we describe its application to a specific augmented reality system.
- We show how data from multiple different sensors can be fused, taking into account the uncertainties associated with each, to yield an improved object-to-HMD pose. In particular, it is shown that a hybrid sensing system combining both head-mounted and fixed sensors can improve registration accuracy over that from either sensor used alone.
- We demonstrate mathematically some insights regarding the characteristics of registration sensors. In particular, we show that the directions of greatest uncertainty for a head-mounted and fixed sensor are nearly orthogonal and that these can be fused in a simple way to improve the overall accuracy.

The remainder of this paper is organized as follows: Section 2 provides a background on pose estimation, with a description of the terminology used in the paper. Section 3 develops the method for estimating the uncertainty of a pose, transforming it from one coordinate frame to another, and fusing two pose estimates. Section 4 describes the particular experimental augmented reality system that was used to test the registration method—that of a surgical aid. Section 5 illustrates the application of the method to the surgical aid system. A typical configuration is analyzed and the predicted accuracy of the combined (hybrid) pose estimate is found to be much improved over that obtained by either sensor alone. Finally, Section 6 provides a discussion.

2 Background on Pose Estimation

2.1 Representation of Pose

The pose of a rigid body {A} with respect to another coordinate system {B} can be represented by a six element vector ${}^B_A\mathbf{x} = ({}^Bx_{Aorg}, {}^By_{Aorg}, {}^Bz_{Aorg}, \alpha, \beta, \gamma)^T$, where ${}^B\mathbf{p}_{Aorg} = ({}^Bx_{Aorg}, {}^By_{Aorg}, {}^Bz_{Aorg})^T$ is the origin of frame {A} in frame {B}, and (α, β, γ) are the angles of rotation of {A} about the (z, y, x) axes of {B}. An alternative representation of orientation is to use three elements of a quaternion; the conversion between Euler angles and quaternions is straightforward [35].

Equivalently, pose can be represented by a 4 × 4 homogeneous transformation matrix [35]:

$$ {}^B_A\mathbf{H} = \begin{pmatrix} {}^B_A\mathbf{R} & {}^B\mathbf{p}_{Aorg} \\ 0 & 1 \end{pmatrix}, \tag{1} $$

where ${}^B_A\mathbf{R}$ is the 3 × 3 rotation matrix corresponding to the angles (α, β, γ). In this paper, we shall use the letter $\mathbf{x}$ to designate a six-element pose vector and the letter $\mathbf{H}$ to designate the equivalent 4 × 4 homogeneous transformation matrix.

Homogeneous transformations are a convenient and elegant representation. Given a homogeneous point ${}^A\mathbf{p} = ({}^Ax_P, {}^Ay_P, {}^Az_P, 1)^T$, represented in coordinate system {A}, it may be transformed to coordinate system {B} with a simple matrix multiplication ${}^B\mathbf{p} = {}^B_A\mathbf{H}{}^A\mathbf{p}$. The homogeneous matrix representing the pose of frame {B} with respect to frame {A} is just the inverse of the pose of {A} with respect to {B}, i.e., ${}^A_B\mathbf{H} = {}^B_A\mathbf{H}^{-1}$. Finally, if we know the pose of {A} with respect to {B} and the pose of {B} with respect to {C}, then the pose of {A} with respect to {C} is easily given by the matrix multiplication ${}^C_A\mathbf{H} = {}^C_B\mathbf{H}{}^B_A\mathbf{H}$.

2.2 Pose Estimation Algorithms

The 2D-to-3D pose estimation problem is to determine the pose of a rigid body, given an image from a single camera (this is also called the "exterior orientation" problem in photogrammetry). Specifically, we are given a set of 3D known points on the object (in the coordinate frame of the object) and the corresponding set of 2D measured image points from the camera, which are the perspective projections of the 3D points. The internal parameters of the camera (focal length, principal point, etc.) are known. The goal is to find the pose of the object with respect to the camera, ${}^{cam}_{obj}\mathbf{x}$. There are many solutions to the problem; in this work, we used the algorithm described by Haralick and Shapiro [16], which uses an iterative nonlinear least squares method. The algorithm effectively minimizes the squared error between the measured 2D point locations and the predicted 2D point locations.

The 3D-to-3D pose estimation problem is to determine the pose of a rigid body, given a set of 3D point measurements[1] (this is also called the "absolute orientation" problem in photogrammetry). Specifically, we are given a set of 3D known points on the object $\{{}^{obj}\mathbf{p}_i\}$ and the

1. These 3D point measurements may have been obtained from a previous triangulation process using a stereo vision sensor.

corresponding set of 3D measured points from the sensor $\{^{sen}\mathbf{p}_i\}$. The goal is to find the pose of the object with respect to the sensor, $^{sen}_{obj}\mathbf{x}$. There are many solutions to the problem; in this work we used the solution by Horn [36], which uses a quaternion-based method.[2] The algorithm effectively minimizes the squared error between the measured 3D point locations and the predicted 3D point locations.

3 Determination and Manipulation of Pose Uncertainty

Given that we have estimated the pose of an object using one of the methods above, what is the uncertainty of the pose estimate? We can represent the uncertainty of a six-element pose vector $\mathbf{x}$, by a 6×6 covariance matrix $\mathbf{C_x} = E(\Delta\mathbf{x}\Delta\mathbf{x}^T)$, which is the expectation of the square of the difference between the estimate and the true vector.

This section describes methods to estimate the covariance matrix of a pose, given the estimated uncertainties in the measurements, transform the covariance matrix from one coordinate frame to another, and combine two pose estimates.

3.1 Computation of Covariance

Assume that we have n measured data points from the sensor $\{\mathbf{p}_i\}$ and the corresponding points on the object $\{\mathbf{q}_i\}$. The object points $\mathbf{q}_i$ are 3D; the data points $\mathbf{p}_i$ are either 3D (in the case of 3D-to-3D pose estimation) or 2D (in the case of 2D-to-3D pose estimation). We assume that the noise in each measured data point is independent and that the noise distribution of each point is given by a covariance matrix $\mathbf{C_p}$.

Let $\mathbf{p}_i = \mathbf{g}(\mathbf{q}_i, \mathbf{x})$ be the function which transforms object points into measured data points for a hypothesized pose $\mathbf{x}$. In the case of 3D-to-3D pose estimation, this is just a multiplication of $\mathbf{q}_i$ by the corresponding homogeneous transformation matrix. In the case of 2D-to-3D pose estimation, the function is composed of a transformation followed by a perspective projection. The pose estimation algorithms described above solve for $\mathbf{x}_{est}$ by minimizing the sum of the squared errors. Assume that have we solved for $\mathbf{x}_{est}$ using the appropriate algorithm (i.e., 2D-to-3D or 3D-to-3D). We then linearize the equation about the estimated solution $\mathbf{x}_{est}$:

$$\mathbf{p}_i + \Delta\mathbf{p}_i = \mathbf{g}(\mathbf{q}_i, \mathbf{x}_{est} + \Delta\mathbf{x}) \approx \mathbf{g}(\mathbf{q}_i, \mathbf{x}_{est}) + \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right]^T_{\mathbf{q}_i, \mathbf{x}_{est}} \Delta\mathbf{x}. \tag{2}$$

Since $\mathbf{p}_i \approx \mathbf{g}(\mathbf{q}_i, \mathbf{x}_{est})$, the equation reduces to

$$\Delta\mathbf{p}_i = \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right]^T_{\mathbf{q}_i, \mathbf{x}_{est}} \Delta\mathbf{x} = \mathbf{M}_i \Delta\mathbf{x}, \tag{3}$$

where $\mathbf{M}_i$ is the Jacobian of $\mathbf{g}$, evaluated at $(\mathbf{q}_i, \mathbf{x}_{est})$. Combining all the measurement equations:

$$\begin{pmatrix} \Delta\mathbf{p}_1 \\ \vdots \\ \Delta\mathbf{p}_n \end{pmatrix} = \begin{pmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_n \end{pmatrix} \Delta\mathbf{x} \Rightarrow \Delta\mathbf{P} = \mathbf{M}\Delta\mathbf{x}. \tag{4}$$

Solving for $\Delta\mathbf{x}$ in a least squares sense, we get $\Delta\mathbf{x} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\Delta\mathbf{P}$. The covariance matrix of $\mathbf{x}$ is given by the expectation of the outer product:

$$\begin{aligned} \mathbf{C_x} &= E(\Delta\mathbf{x}\,\Delta\mathbf{x}^T) \\ &= E\left[(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\Delta\mathbf{P}\Delta\mathbf{P}^T\left((\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\right)^T\right] \\ &= (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T E(\Delta\mathbf{P}\Delta\mathbf{P}^T)\left((\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\right)^T \\ &= (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T \begin{pmatrix} \mathbf{C_p} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{C_p} \end{pmatrix} \left((\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\right)^T. \end{aligned} \tag{5}$$

Note that we have assumed that the errors in the data points are independent, i.e., $E(\Delta\mathbf{p}_i\Delta\mathbf{p}_j^T) = 0$, for $i \neq j$. If the errors in different data points are actually correlated, our simplified assumption could result in an underestimate of the actual covariance matrix. Also, the above analysis was derived assuming that the noise is small. However, we computed the covariance matrices for the configuration described in Section 4, using both (5) and using a Monte Carlo simulation, and found (5) is fairly accurate even for noise levels much larger than in our application. For example, using input noise with variance 225 mm^2 (compared to the actual 0.0225 mm^2 in our application) the largest deviation between the variances of the translational dimensions was 5.5 mm^2 (out of 83 mm^2).

3.2 Transformation of Covariance

We can transform a covariance matrix from one coordinate frame to another. Assume that we have a six-element pose vector $\mathbf{x}$ and its associated covariance matrix $\mathbf{C_x}$. Assume that we apply a transformation, represented by a six-element vector $\mathbf{w}$, to $\mathbf{x}$ to create a new pose $\mathbf{y}$. Denote $\mathbf{y} = \mathbf{g(x, w)}$. A Taylor series expansion yields $\Delta\mathbf{y} = \mathbf{J}\Delta\mathbf{x}$, where $\mathbf{J} = (\partial\mathbf{g}/\partial\mathbf{x})$. The covariance matrix $\mathbf{C_y}$ is found by:

$$\begin{aligned} \mathbf{C_y} &= E(\Delta\mathbf{y}\Delta\mathbf{y}^T) = E\left[(\mathbf{J}\Delta\mathbf{x})(\mathbf{J}\Delta\mathbf{x})^T\right] \\ &= \mathbf{J}E(\Delta\mathbf{x}\Delta\mathbf{x}^T)\mathbf{J}^T = \mathbf{J}\mathbf{C_x}\mathbf{J}^T. \end{aligned} \tag{6}$$

A variation on this method is to assume that the transformation $\mathbf{w}$ also has an associated covariance matrix $\mathbf{C_w}$. In this case, the covariance matrix $\mathbf{C_y}$ is:

$$\mathbf{C_y} = \mathbf{J_x}\mathbf{C_x}\mathbf{J_x}^T + \mathbf{J_w}\mathbf{C_w}\mathbf{J_w}^T, \tag{7}$$

where $\mathbf{J_x} = (\partial\mathbf{g}/\partial\mathbf{x})$ and $\mathbf{J_w} = (\partial\mathbf{g}/\partial\mathbf{w})$. The above analysis was verified with Monte Carlo simulations, using both the 3D-to-3D algorithm and the 2D-to-3D algorithm.

3.3 Interpretation of Covariance

A useful interpretation of the covariance matrix is obtained by assuming that the errors are jointly Gaussian. The joint probability density for n-dimensional error vector $\Delta\mathbf{x}$ is [37]:

2. This is the algorithm used in the Northern Digital Optotrak sensor, described in Section 4.

$$p(\Delta\mathbf{x}) = \left(|2\pi|^{N/2}|\mathbf{C_x}|^{1/2}\right)^{-1}\exp\left(-\tfrac{1}{2}\Delta\mathbf{x}^T\mathbf{C_x^{-1}}\Delta\mathbf{x}\right). \quad (8)$$

If we look at surfaces of constant probability, the argument of the exponent is a constant, given by the relation $\Delta\mathbf{x}^T\mathbf{C_x^{-1}}\Delta\mathbf{x} = z^2$. This is the equation of an ellipsoid in n dimensions. For a given value of z, the cumulative probability of an error vector being inside the ellipsoid is P. For $n = 3$ dimensions, the ellipsoid defined by $z = 3$ corresponds to a cumulative probability P of approximately 97 percent.[3]

For a six-dimensional pose $\mathbf{x}$, the covariance matrix $\mathbf{C_x}$ is 6×6 and the corresponding ellipsoid is six-dimensional (which is difficult to visualize). However, we can select only the 3D translational component of the pose and look at the covariance matrix corresponding to it. Specifically, let $\mathbf{z} = (x, y, z)^T$ be the translational portion of the pose vector $\mathbf{x} = (x, y, z, \alpha, \beta, \gamma)^T$. We obtain $\mathbf{z}$ from $\mathbf{x}$ using the equation $\mathbf{z} = \mathbf{M}\,\mathbf{x}$, where $\mathbf{M}$ is the matrix

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (9)$$

The covariance matrix for $\mathbf{z}$ is given by $\mathbf{C_z} = \mathbf{M}\,\mathbf{C_x}\,\mathbf{M}^T$ (which is just the upper left 3×3 submatrix of $\mathbf{C_x}$). We can then visualize the uncertainty in position using the three-dimensional ellipsoid corresponding to the set $\{\mathbf{z}|(\mathbf{z}-\bar{\mathbf{z}})^{\mathbf{T}}\mathbf{C_z^{-1}}(\mathbf{z}-\bar{\mathbf{z}}) \le 9\}$.

We can visualize the uncertainty in the rotational component of the pose by finding the uncertainties in the directions of the x, y, z axes of the coordinate frame relative to the world frame. The orientation of a particular axis $\mathbf{a}$ of the coordinate frame is found using $\mathbf{a} = \mathbf{R}(\alpha, \beta, \gamma)\mathbf{e}$, where $\mathbf{R}(\alpha, \beta, \gamma)$ is the rotation matrix of the coordinate frame in the world and $\mathbf{e}$ is the relevant unit vector in the world frame. Using the results of the previous section, the covariance of $\mathbf{a}$ is given by $\mathbf{C_a} = \frac{\partial}{\partial\hat{\mathbf{e}}}[\mathbf{R}(\alpha,\beta,\gamma)\mathbf{e}]\mathbf{C}_{\hat{\mathbf{e}}}\frac{\partial}{\partial\hat{\mathbf{e}}}[\mathbf{R}(\alpha,\beta,\gamma)\mathbf{e}]^T$, where $\mathbf{C}_{\hat{\mathbf{e}}}$ is the 3×3 lower right submatrix of $\mathbf{C_x}$ corresponding to the angular uncertainty and $\frac{\partial}{\partial\hat{\mathbf{e}}}[\mathbf{R}(\alpha,\beta,\gamma)\mathbf{e}]$ is the Jacobian of $\mathbf{R}(\alpha, \beta, \gamma)\mathbf{e}$ with respect to α, β, γ. $\mathbf{C_a}$ is of rank 2 and the ellipsoid associated with it will be "flat" in the direction perpendicular to $\mathbf{a}$. For visualization, these ellipsoids define the bases of cones drawn about each axis and show how the ends of the axis would move given the variation in the Euler angles.

To illustrate these concepts, a simulation of a pose estimation process was performed. A simulated target pattern was created, attached to a coordinate frame A. The pose of coordinate frame A with respect to a sensor S, ${}^S_A\mathbf{H}$, was estimated using a 3D-to-3D algorithm. The covariance matrix of the resulting pose, $\mathbf{C_A}$, was computed using (5). Fig. 1 shows a rendering of the ellipsoid corresponding to the uncertainty of the translational component of the pose. The ellipsoid is shown centered at the origin of frame A. The rotational uncertainty is depicted as elongated cones about each axis. Note that, although the ellipsoid (representing the translational uncertainty) is almost spherical, the cones (representing the rotational uncertainty) are asymmetrical. The uncertainty is greatest for rotations about the long axis of the target pattern and, so, the cones perpendicular to that axis are elongated. This is because the shorter dimension of the target pattern provides less orientation constraint than the longer dimension.

3. The exact formula for the cumulative probability in N dimensions is $1 - P = \frac{N}{2^{N/2}\Gamma(N/2+1)}\int_z^\infty X^{N-1}e^{-X^2/2}dX$ [37].

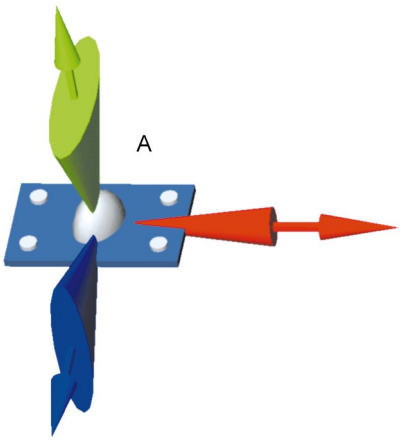

Fig. 1. A visualization of the uncertainty of the pose of a coordinate frame. The ellipsoid, shown centered at the origin of the coordinate frame, represents the uncertainty in the translational component of the pose. The rotational uncertainty is depicted as elongated cones about each axis.

To illustrate the effect of transformations on covariance matrices, another simulation of a pose estimation process was performed. The target pattern used in Fig. 1 was attached to coordinate frame A and the uncertainty of the pose of A with respect to sensor S was computed. As shown in Fig. 2, the translational component of the uncertainty is represented by a small ellipsoid centered at A and the rotational component of the uncertainty is represented by elongated cones about each axis of A. Next, two other objects with coordinate frames B and C were rigidly attached to A, at known poses with respect to A. The poses of B and C with respect to S were derived via ${}^S_B\mathbf{H} = {}^S_A\mathbf{H}\,{}^A_B\mathbf{H}$ and ${}^S_C\mathbf{H} = {}^S_A\mathbf{H}\,{}^A_C\mathbf{H}$, respectively. The covariance matrices of these poses, $\mathbf{C_B}$ and $\mathbf{C_C}$, were then estimated using (6). The uncertainties of the translational components of $\mathbf{C_B}$ and $\mathbf{C_C}$ are shown by the ellipsoids centered at B and C, respectively.

Note that the ellipsoids for $\mathbf{C_B}$ and $\mathbf{C_C}$ are much larger than the ellipsoid for $\mathbf{C_A}$, even though the relative poses of B and C with respect to A are known exactly. This is due to the orientation uncertainty in the pose of A with respect to S, which gives rise to an uncertainty in the location of B and C. The uncertainty is greatest in the plane perpendicular to the line to object A—hence, the flattened shapes of the ellipsoids associated with $\mathbf{C_B}$ and $\mathbf{C_C}$. Note that the shape of the flattened ellipsoids corresponds to the shape of the cones about the axes perpendicular to the flattened parts.

In general, the component of translational uncertainty in a frame B that is caused by the orientation error in A can be estimated by $\Delta P = d\,\Delta\theta$, where $\Delta\theta$ is the orientation error and d is the distance between A and B. Thus, the uncertainty in the derived location of B grows with the orientation uncertainty in A and also with the distance between A and B. If one needs to track an object using a

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.