A General Method for Comparing the Expected Performance of Tracking and Motion Capture Systems

B. Danette Allen[†‡] and Greg Welch[‡]

[†]NASA Langley Research Center
Hampton, VA 23681, U.S.A
danette.allen@nasa.gov

[‡]University of North Carolina at Chapel Hill
Chapel Hill,NC 27599-3175, U.S.A.
{welch, bdallen}@cs.unc.edu

ABSTRACT

We introduce a general method for evaluating and comparing the expected performance of sensing systems for interactive computer graphics. Example applications include head tracking systems for virtual environments, motion capture systems for movies, and even multi-camera 3D vision systems for image-based visual hulls.

Our approach is to estimate the asymptotic position and/or orientation uncertainty at many points throughout the desired working volume, and to visualize the results graphically. This global performance estimation can provide both a quantitative assessment of the expected performance, and intuition about the type and arrangement of sources and sensors, in the context of the desired working volume and expected scene dynamics.

Categories and Subject Descriptors

I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism — Virtual reality; I.4.4 [**Image Processing and Computer Vision**]: Restoration — Kalman filtering; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis — Sensor fusion; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis — Motion; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis — Tracking; G.3 [**Probability and Statistics**]: Stochastic processes

General Terms

Measurement, Performance, Design

Keywords

virtual environments, tracking, motion capture, sensor fusion, covariance analysis, information visualization, computer vision

VRST'05, November 7–9, 2005, Monterey, California, USA.

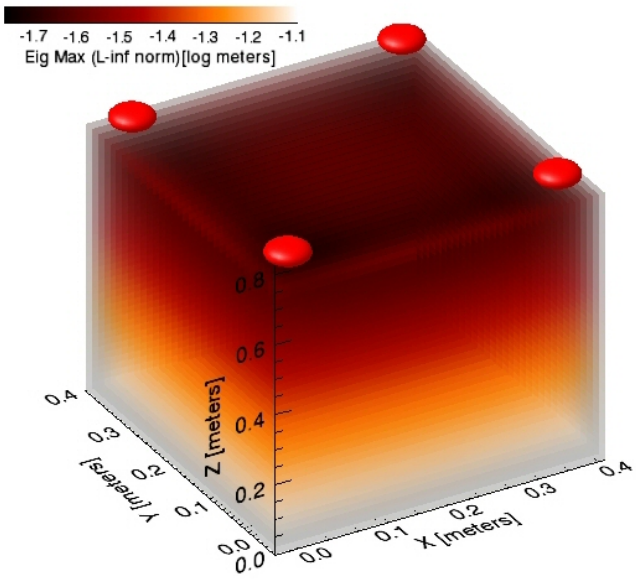

Figure 1: An example showing global steady-state estimation to evaluate a simple range-based acoustic tracking system. The four red spheres at the top represent acoustic transmitters, and the volume visualization depicts the fundamental uncertainty in the estimated position of an acoustic sensor throughout the 0.4×0.4 meter working volume. The darker areas near the transmitters reflect lower uncertainty (better expected performance), while the brighter areas near the floor indicate greater uncertainty (worse expected performance).

1. INTRODUCTION

By definition, *interactive* computer graphics applications include some system for estimating the position and/or orientation of real targets over time. Typical examples include head tracking systems for virtual environments and motion capture systems for movies. While the estimation circumstances and performance requirements vary with each application, the fundamental source of information is the same: the estimates are derived from electrical measurements of devices such as mechanical, inertial, optical, acoustic, and magnetic source/sensors [31]. Each type of device has fundamental limitations related to the physical medium, practical limitations im-

posed by the measurement systems, and application-specific limitations related to the motion characteristics of the target being observed. These limitations affect the quantity and quality of the information throughout the desired working volume for a given system configuration, and they do so in a complex and often unclear way. As such, despite the best practices of experienced engineers, many design choices end up being based on subtle intuition developed over many years, as opposed to specific, quantifiable design assessments. And once a system appears to work, people hesitate to change it, for fear of negatively impacting performance in some unforeseen manner.

Despite (or because of) these difficulties, tracking and motion capture for interactive computer graphics have been explored for over 35 years [26]. Complete historical surveys include [22, 4, 20, 31]. Commercial and research teams have explored mechanical, magnetic, acoustic, inertial, and optical technologies. For example, commercial magnetic tracking systems from Ascension and Polhemus have historically enjoyed popularity as a result of a small user-worn component, relative ease of use, and robustness for many applications. Optical systems include the HiBall-3000^{TM} system by 3rdTech, the FlashPoint and Pixsys^{TM} systems by Image Guided Technologies, and the laserBIRD^{TM} system by Ascension Technology. Foxlin et al. at Intersense in particular have had tremendous success developing hybrid systems that combine inertial measurements with acoustic signals [10, 11, 12], and more recently with passive optical signals [10]. Similarly optical systems for 3D motion capture have a long history, having been explored for over 30 years [32]. Today companies like Vicon, Motion Analysis, and Ascension make turn-key optical systems that are used, for example, in human and animal motion analysis, movies, and industrial applications. Much work has also been done on vision-based approaches to motion capture [21].

Yet many users and designers would like to reduce or better configure their tracking and motion capture infrastructure, while maintaining or even improving the level of performance. The problem is that both reduction and reconfiguration of such systems can be very difficult, as complex interactions or dependencies between devices are difficult to understand. For example, what would be the likely effect of removing or re-arranging a tracking system's optical or acoustic beacons? How and where will the addition of nearby light or sound-occluding objects affect the performance? How will moving or redirecting one motion capture camera affect the precision? Will it help to add another camera? How many do you need? What happens if you change the lenses?

1.1 System Design

Historically, designers have relied on educated intuition and experience when making design choices. As illustrated on the left side of Figure 2, they begin with requirements for working volume, expected user motion (dynamics), and infrastructure. These requirements are coupled with a candidate design(s) that includes a tracking medium (or hybrid combination of mediums), associated source/sensor devices (hardware), and some algorithm (software) for combining the information from the devices. Once the algorithm is chosen, a designer can run simulations (using any method of choice) and build prototypes for in-situ testing. In this way, an estimate of system performance can be found. However, simulations and prototyping rely on a specific algorithm, a specific motion path and, in the case of prototyping, demand a working system.

Long before algorithm selection and hardware fabrication, it is the choice and configuration of the source/sensor devices that are critical for most systems. No estimation algorithm can overcome poor choices of devices, parameters, or geometric arrangement. If

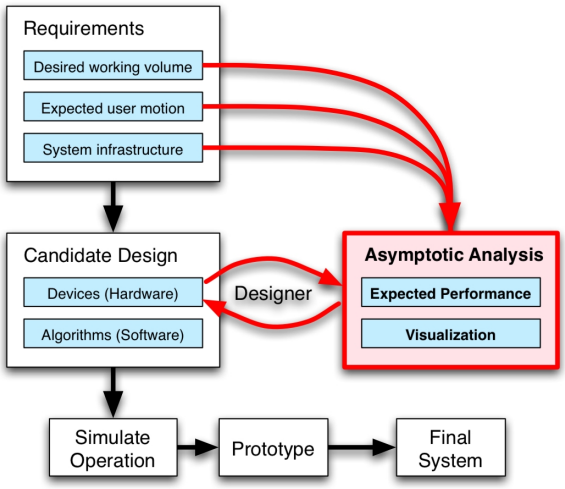

Figure 2: The tracking system design process showing the historical process on the left with the proposed method on the right (in red) for interaction with hardware choices.

the necessary information is not available at a sufficient rate and quality throughout the desired working volume, the performance is inherently limited in those areas. In effect the device choices set an upper bound on how well the system as a whole will perform.

Ideally one could specify desired performance goals throughout the working volume, and have a computer search the entire space and present the optimal design. However for all but the most trivial systems the design space is so large as to render the search intractable, making automatic optimization impractical if not impossible, except in relatively restricted circumstances [23]. There is work available that addresses efficient allocation of cameras for tracking systems [7] and accuracy prediction for marker-based tracking [9] but this research is restricted strictly to camera-based systems in specific contexts.

We believe a useful alternative to such *artificial intelligence* is the notion of *intelligence amplification* [3, 2]. The idea is to develop methods and tools that provide a human with insight into variations in the expected performance throughout the desired working volume for a *particular* design choice, as well as the relative global effects of variations *between* candidate designs, independent of the tracking algorithm chosen for the real system.

1.2 Our Approach

Specifically our goal has been to develop general methods and tools to allow one to interact with a candidate hardware system, varying the types and configurations of devices and graphically visualizing the corresponding effects on the global performance. This value-added step in the design process is shown on the right (in red) in Figure 2. Similar to fluid or air-flow visualizations, our goal is to make "invisible" sensor information "visible" throughout a working volume, so that a designer can develop insights into the effects of their design choices. While such visualizations have been done before for specific systems [11, 18, 25, 17], we present a general framework that will accommodate virtually any tracking or motion capture system, including multi-camera setups for image-based visual hulls [19].

Our approach is to use a stochastic framework for estimating the asymptotic performance of a candidate design, and then to produce surface or volumetric visualizations of the results. The stochastic framework integrates descriptions of the devices, the bounds of the

desired working volume, and a model for the expected user motion, to produce a graphical depiction of the expected position and/or orientation uncertainty throughout the working volume as illustrated in Figure 1. One can then alter device parameters and repeat the visualization, to see how it affects the expected performance.

Our goal with this paper is to convey the fundamental concepts, and to explain the general approach so that motivated readers will be able to use the techniques on their own to evaluate (for example) alternative tracking, motion capture, and image-based visual hull setups. Ultimately we hope to develop and make available tools and models that might transform the way people approach the typical design process, putting what is often times an ad hoc process on more solid theoretical foundations, and giving people the tools to explore what is otherwise an intractable design space.

1.3 This Paper

In Section 2 we describe the basic approach and the specific mathematical framework we use to quantify the global uncertainty corresponding to a candidate design. In Section 3 we present some experiments aimed at validating the use of asymptotic estimates and visualizations, and some concrete examples of the approach being used to evaluate other systems. Finally in Section 4 we summarize and discuss our future plans for the work.

Throughout the paper we use lower-case variables with over-bars to denote a vector and upper-case variables to denote matrices. We use the term *designer* to refer to the engineer or researcher evaluating the system, and the term *target* to refer to the object actually being tracked/captured. Example targets include a sensor on a person's head, a retroreflective sphere on a joint or limb, and potential 3D surface points that one wants to reconstruct using cameras and image/vision-based techniques.

Finally, throughout the paper we use a simple acoustic 3D position tracking system to provide a concrete basis for discussion. (In Section 3 we look at more realistic systems.) This acoustic system is depicted in Figure 3, and a corresponding visualization of the estimated performance is shown in Figure 1. There are four speakers permanently mounted in the corners of the ceiling, and a microphone (the tracking target) mounted on the moving user. The curve in the middle represents an example target motion path through the 3D space over time, and the point $\bar{x}(t) \in \Re^3$ represents the 3D position or *state* of the target at time t.

2. PERFORMANCE ESTIMATION

There are many possible quantitative metrics for performance estimation. For example one might be concerned about resolution or precision, noise, static accuracy, dynamic accuracy, latency, or some combination. See [1, 6, 15] for more examples and general discussion of performance.

Our approach to is to use a stochastic estimate of the asymptotic or *steady-state* error covariance throughout the working volume. Consider our example acoustic 3D position tracking system. At a representative set of 3D points $\{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_p\}$ throughout the working volume we can estimate and graphically depict

$$P_i^{\infty} = \lim_{t \to \infty} \mathsf{E}\left\{ (\bar{x}_i(t) - \tilde{x}_i(t))(\bar{x}_i(t) - \tilde{x}_i(t))^{\mathsf{T}} \right\}, 1 \le i \le p \qquad (1)$$

where $\bar{x}_i$ and $\tilde{x}_i$ represent the *true* and *estimated* states (respectively) at point i, and E denotes statistical expectation. Note that we do not actually attempt to estimate $\bar{x}_i$ or $\tilde{x}_i$. Instead we estimate P^{∞} directly from state-space models of the system and stochastic estimates of the various noise sources, as described below in the remainder of this section.

A stochastic steady-state approach is attractive for many reasons.

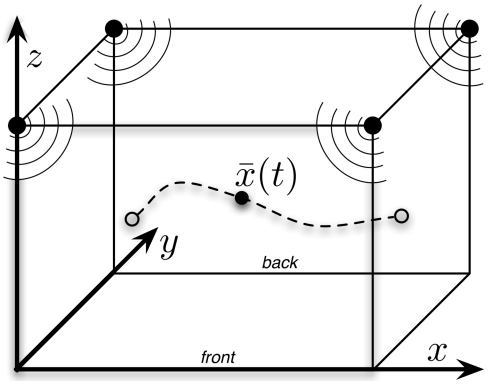

Figure 3: A simple acoustic 3D position tracker example.

For one it nicely accounts for measurement noise, sampling rates, measurement sensitivities, and expected target motion dynamics. Furthermore virtually any tracking or motion capture system can be described in a stochastic form[1], and there are relatively well-understood methods for estimating the corresponding steady-state performance. And while the absolute accuracy of the uncertainty estimates will depend on several factors, including the accuracy of the input and models provided to the system, we believe that most designers would likely have (or have access to) reasonably appropriate noise and device models.

2.1 System Models

To estimate the steady-state error covariance we being by mathematically describing the measurement system and the expected target motion. We do so using state-space models [14] that will look familiar to anyone acquainted with the Kalman filter [16, 27]. In fact it might at first appear that we are describing a Kalman filter-based tracking algorithm as in [11] or [28]. However the mathematics of steady-state estimation conceptually reverse the signal direction we would normally think about in a tracking or motion capture system. Normally we would think about propagating a signal from a point in the working volume, through a noisy sensor, and into an estimation algorithm. Instead the steady-state estimate in effect propagates estimated measurement noise signals through models of the measurement systems, back into the working volume (the state space), combines them with expectations for the target motion dynamics, and produces an estimate of the *fundamental* uncertainty (error covariance P^{∞}) at a particular point.

We begin with a general description of the appropriate state-space models, and then provide some concrete examples. Given an n-dimensional state $\bar{x}(t)$, the target motion over time δt can typically be modeled as a first-order dynamic process:

$$\bar{x}(t) = A\bar{x}(t - \delta t) + G\bar{w}(t - \delta t) \qquad (2)$$

where A is an $n \times n$ *state transition matrix*, $\bar{w}$ is a zero-mean and spectrally white $n \times 1$ random signal known as the *process noise*, and G is an $n \times n$ noise shaping matrix. In addition it is common to model the m-dimensional device measurements $\bar{z}$ at time t as

$$\bar{z}(t) = H\bar{x}(t) + \bar{v}(t) \qquad (3)$$

where H is an $m \times n$ matrix relating the n-dimensional state to the m-dimensional measurements, and $\bar{v}$ represents zero-mean, white measurement noise, presumed to be uncorrelated with $\bar{w}$.

[1] Such stochastic frameworks make assumptions about noise characteristics, etc. Even if these assumptions are not completely accurate they are often adopted for simplicity or lack of a better model.

Equations (2) and (3) are often referred to respectively as the *process model* and the *measurement model*, and they serve in some form as the basis for most stochastic estimation methods. In the case of tracking, the former models the target's motion over time, while the latter models the outputs of the devices (sensors) used to estimate the target's position.

In practice the actual noise signals $\bar{w}$ and $\bar{v}$ are not known or estimated as part of a stochastic estimator. Instead designers typically model the process and measurements as

$$\bar{x}(t) = A\bar{x}(t-\delta t), \quad (4)$$
$$\bar{z}(t) = H\bar{x}(t), \quad (5)$$

then estimate the process and measurement noise covariances Q and R of the presumed normal distributions $\bar{w} \sim \mathsf{N}(0,Q)$ and $\bar{v} \sim \mathsf{N}(0,R)$, and use those covariances to weight the measurements and to estimate the state uncertainty. It is the parameters A, H, Q and R (as described below) that the designer must "know" (specify) to perform a steady-state analysis.

In cases where the process and/or measurement models are non-linear, equations (4) and (5) would be written as

$$\bar{x}(t) = f(\bar{x}(t-\delta t)), \quad (6)$$
$$\bar{z}(t) = h(\bar{x}(t)). \quad (7)$$

These non-linear functions can be linearized about the point of interest $\bar{x}$ in the state space. To do so one would compute the Jacobians of the respective functions,

$$A = \frac{\partial}{\partial \bar{x}} \bar{f}(\bar{x})\Big|_{\bar{x}} \quad (8)$$
$$H = \frac{\partial}{\partial \bar{x}} \bar{h}(\bar{x})\Big|_{\bar{x}} \quad (9)$$

and use them in place of their corresponding matrices in equations (4) and (5). In fact this is what we typically do. While such linearizations can lead to sub-optimal results, they provide a computationally efficient means for estimation, and in most cases should offer a reasonable basis for comparison of steady-state results.[2] For linear models, the designer would write functions that implement A and H (linear functions in matrix form) from equations (4) and (5). For non-linear models, the designer would instead write functions that implement the respective Jacobians from equations (8) and (9).

2.1.1 The Process Model

In the process model described by Equation (2), the state transition matrix A and noise shaping matrix G each play a role in moving the user's state forward over some (typically small) interval of time. The term $A\bar{x}$ models the deterministic portion of the process, while the term $G\bar{w}$ and corresponding covariance Q model the random portion of the user's motion.[3]

Here we use our acoustic tracker example to provide a more concrete notion of the process model parameters A and Q. To begin with, let us expand our 3D state $\bar{x}$ introduced earlier to include the target position and *derivatives* (velocities) of the target. The six-dimensional state would then be

$$\bar{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^{\mathsf{T}}. \quad (10)$$

To move the single element x of the state vector $\bar{x}$ forward over time δt one would compute the new position x as a function of the previous value $x(t-\delta t)$, the corresponding velocity element $\dot{x}$, and the time δt since the last update: $x(t) = x(t-\delta t) + \dot{x}(t-\delta t) \times \delta t$. The complete corresponding state transition matrix A, which is actually a function of δt, would be

$$A = \begin{bmatrix} 1 & 0 & 0 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11)$$

Now consider the random component of the process given by Equation (2). The process noise $\bar{w}$ is presumed to be a normally-distributed, zero-mean, spectrally white random variable with distribution $\bar{w} \sim \mathsf{N}(0,Q)$. If we assume the process noise "flows" through (is shaped by) the same system of integrators represented by Equation (11), then the covariance Q can be described as

$$Q[i,i] = q\frac{(\delta t)^3}{3} \quad (12)$$
$$Q[i,j] = Q(\delta t)[j,i] = q\frac{(\delta t)^2}{2} \quad (13)$$
$$Q[j,j] = q\delta t \quad (14)$$

for each pair $(i,j) \in \{(1,4),(2,5),(3,6)\}$ and some noise magnitude q. The above derivation of Q can be found in [5], and discussion about choosing q can be found in [29].

It is worth noting here that while one might imagine the need for many different process (target motion) models, our experience indicates that the above position-velocity model is a reasonable match for the average human motion. If one expected the target to be primarily still, one might want to eliminate the velocity states in $\bar{x}$. Similarly if one expected the target to undergo coherent accelerations, one could add acceleration states.

2.1.2 The Measurement Model

In the measurement model described by Equation (3) the *measurement matrix* H determines the relationship between the state and the measurements. Each type of device (combination sensor and/or source) would typically require a different measurement model. In our acoustic example (see Figure 3) there are four speakers fixed in the environment, and the target is a single moving microphone. Our "candidate" example tracking system will continually measure the range from the microphone to each of the four speakers using a time-of-flight approach. In this case $m = 4$ and the measurement function would be

$$\bar{z}[i] = \bar{h}_i(\bar{x}) = \sqrt{(\bar{x}[x] - \bar{t}[i,x])^2 + (\bar{x}[y] - \bar{t}[i,y])^2 + (\bar{x}[z] - \bar{t}[i,z])^2}$$

for each transmitter $1 \le i \le 4$, where $\bar{t}[i,*]$ represents the position of transmitter i. Because our acoustic system uses four scalar range measurements and its state vector is six-dimensional, the measurement matrix H must be a 4×6 matrix. In fact because our measurement function is non-linear, we would have to use the linear approximation given by the measurement Jacobian as in Equation (7). For example, the Jacobian element corresponding to transmitter number one and the x element of the state would be

$$H_{1,x} = \frac{\bar{x}_x - \bar{t}_{1,x}}{\sqrt{(\bar{x}[x] - \bar{t}[i,x])^2 + (\bar{x}[y] - \bar{t}[i,y])^2 + (\bar{x}[z] - \bar{t}[i,z])^2}}.$$

Referring back to Equation (3), the measurement noise $\bar{v}$ is a normally-distributed, zero-mean, spectrally white, random variable with probability distribution $\bar{v} \sim \mathsf{N}(0,R)$. The magnitude of the covariance R

[2] If there is concern, one can carry out a separate analysis of the likely linearization error.

[3] Here and in subsequent expressions we will omit the time parameters t and δt from variables when possible to reduce complexity.

represents the expected measurement noise for the given combination of sources and sensors. Unlike the process noise q in in (12)–(14), the measurement noise has concrete origins, and in practice R can be estimated with relative ease. For example, one can arrange a real source/sensor pair in a lab, and gather statistics on the measurement variance under representative conditions, then later fit a function to those gathered statistics and use this function in the asymptotic analysis. Or one can simply estimate the form and magnitude of the noise based on past experience or simulation as in [8]. For our acoustic tracker example, R is the expected variance in the range measurements, which is a function of the range itself.

2.2 Steady-State Solution

The Discrete Algebraic Riccati Equation (DARE) represents a closed-form solution to the steady-state covariance P^∞ [13]. Assuming the process and measurement noise elements are uncorrelated the DARE can be written as

$$P^\infty = APA^\mathsf{T} + Q - AP^\infty H^\mathsf{T}\left(R + HP^\infty H^\mathsf{T}\right)^{-1} HP^\infty A^\mathsf{T}.$$

We use the MacFarlane–Potter–Fath "Eigenstructure Method" [13] to calculate the DARE solution P^∞ as follows. Given the model parameters A, Q, H, and R from Section 2.1 we first calculate the $2n \times 2n$ discrete-time Hamiltonian matrix Ψ as

$$\Psi = \begin{bmatrix} A + QA^{-\mathsf{T}}H^\mathsf{T}R^{-1}H & QA^{-\mathsf{T}} \\ A^{-\mathsf{T}}H^\mathsf{T}R^{-1}H & A^{-\mathsf{T}} \end{bmatrix}. \tag{15}$$

We then form

$$\begin{bmatrix} B \\ C \end{bmatrix} = [\bar{e_1}, \bar{e_2}, \ldots, \bar{e_n}] \tag{16}$$

from the n characteristic eigenvectors $[\bar{e_1}, \bar{e_2}, \ldots, \bar{e_n}]$ of Ψ, and finally using B and C we compute the steady-state covariance as

$$P^\infty = BC^{-1}. \tag{17}$$

As described in the next section we do this at a representative set of points $\{\bar{x_1}, \bar{x_2}, \ldots, \bar{x_p}\}$ throughout the working volume, computing H and R at each point, and using the same A and Q throughout.

Note that one can also compute the *decay time constant* $\tau(A, Q, H, R)$ corresponding to P^∞ [13]. This indicates the time it would take any algorithm to converge on an estimate at $\bar{x_i}$, with steady-state uncertainty P^∞, given the available measurements. This is in some sense a lower bound on the latency of the system at point $\bar{x_i}$.

2.3 Complete Steady-State Computation

In the preceding section we provided a general solution to P^∞. Here we illustrate the process we use to compute P^∞ for all of the desired points in the working volume.

To begin with one has to define the *process model*. In particular one must decide on the form of $A(\delta t)$ and $Q(\delta t)$, for example as in equations (11)–(14). Note that we include the δt parameters here to emphasize that A and Q are functions of δt.

Next one needs to define distinct *measurement models* and corresponding H and R matrices (functions) for each device type. For our example acoustic tracking system, one could think of four separate measurement models or a single parametric model. If using the latter approach, H and R are functions of $\bar{x}$ and any other parameters of interest (e.g., electrical biases, focal length, etc.). In a hybrid system one would typically have multiple parametric measurement models, e.g., one for acoustic devices and one for cameras as in the hybrid example we present in Section 3. In any case the function that implements each measurement model must handle device-specific processing (e.g., beacon selection) and exceptions such as limited fields of view or occlusions. Any measurement model can be defined to test for occlusion and include its effects in its calculation. This is the case with the system described in Section *3.1.1* and, while not included in the analysis presented here, [7] presents a probabilistic model for dynamic self-occlusion that could also be incorporated into a measurement model.

It is important to note that the sample time δt used throughout the paper, for example in $A(\delta t)$ and $Q(\delta t)$, is defined by the *measurement devices*. If the candidate devices provide measurements at 100 Hz, then $\delta t = 0.01$ seconds. If there are multiple devices with different measurement rates, then there are multiple corresponding δt values, with corresponding instances of $A(\delta t)$ and $Q(\delta t)$.

Finally one has to decide at what points $\{\bar{x_1}, \bar{x_2}, \ldots, \bar{x_p}\}$ to evaluate P^∞. One could choose a set of points on a surface in the working volume, or a set that spans some 3D volume, perhaps on a regular 3D grid. We present examples of both in Section 3.

We illustrate the overall process with pseudo-code. Notice how the contributions from each device are similarly fused at every point, no matter what type of device (or combination of devices). This is what makes the approach so general.

```
For each n-dimensional point x̄_i ∈ {x̄_1, x̄_2, ..., x̄_p}
    Ψ_Σ = zeros(2n, 2n)
    For each device
        Determine δt for the device
        Evaluate A with that δt using (11)
        Evaluate Q with that δt using (12)–(14)
        Evaluate H at x̄_i
        Evaluate R at x̄_i
        Compute Ψ using (15)
        Ψ_Σ = Ψ_Σ + Ψ
    Compute the n eigenvectors of Ψ_Σ as in (16)
    Compute P_i^∞ as in (17)
```

Note that the H and R must be evaluated with the appropriate measurement (or Jacobian) function for the device.

Finally one can use surface or volume visualization techniques to render the complete set of points P_i^∞ for $1 \le i \le p$. We show examples of both in the following section.

3. RESULTS

The results presented in this section and throughout this paper are the product of both Matlab™ simulations and our working system prototype, Artemis. Artemis is written in Interactive Data Language™ (IDL), a development environment targeted for data analysis and visualization applications. We chose IDL because of its built-in image processing, matrix math, GUI builder, and cross-platform portability. While Artemis was developed in the Windows environment, IDL's Virtual Machine allows deployment across multiple platforms including Mac OS X and Linux.

3.1 Evaluation

Our approach to evaluating estimated expected performance was twofold. First, we compared our steady-state predictions to the measurement error reported in [30] using a HiBall-3000™ optical tracking system. Second, we performed a series of controlled experiments (again using the HiBall-3000™), estimated the actual performance over a wide area, and compared the results to the steady-state predictions.

Figure 5 shows our experimental setup in which we moved the HiBall sensor along a series of paths mechanically constrained along a rigid rail. 3rdTech engineers graciously provided us with low-

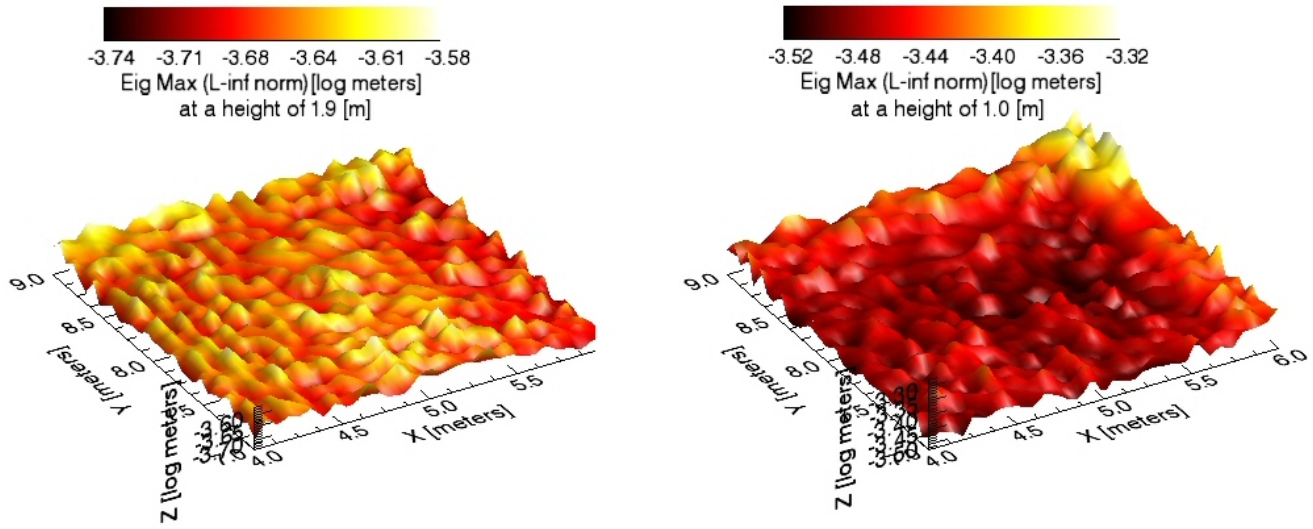

Figure 4: P^{∞} estimates at Left: 1.9 meter "head" height, and Right: 1.0 meter "waist" height

level access to the system, so we were able to log HiBall data at a very high rate (1–3 KHz). We then used the data to compare with steady-state estimates. In particular, we fitted curves to the "rail path" data, then looked at deviations from those curves.

Our experimental rail rig was approximately seven feet (1.4 m) long, with a constant height of approximately 1.9 meters. In order to perform a wide-area comparison, we positioned the rig along 28 paths crisscrossing the area, forming a grid that ranged from 3.0 to 6.0 meters in the x-direction and 5.5 to 10.0 meters in y. We intentionally arranged the grid to extend beyond one ceiling edge by about one meter, allowing for a region of potentially deteriorating performance where the HiBall is starved for measurements.

3.1.1 The HiBall™ System

The HiBall-3000™ system estimates the sensor (target) pose by sighting a two-dimensional array of ceiling-mounted light emitting diodes (LEDs). For our steady-state estimates we modeled the HiBall using the same process model the system uses, and a single 1000 Hz measurement model. In the measurement model, we specified a grid of beacons identical to an actual installation, spaced 10 cm apart across a 6.3 m by 9.0 m ceiling. Just as the actual system continually chooses a set of nearby LEDs for tracking, our measurement model chooses a subset of visible LEDs for performance analysis. Further, both measurement models incorporate occlusion testing in order to choose visible LEDs.

In 1999, the developers of the HiBall tracking system reported estimation errors of 0.2 mm "for nearly all of the working volume" and 0.5 mm at a height of approximately 1 meter [30]. We modeled this setup for comparison and Figure 4 shows the resulting surface plots of P^{∞} analysis at 1.9 meters (head height) and at 1.0 meter (waist height) across an area at the corner of our current HiBall ceiling. At 1.9 meters, the P^{∞} estimates are around 0.20 to 0.22 mm (-3.6 to -3.7 log meters). Note the peaks and valleys in performance corresponding to the location of the LED strips in the ceiling. At first glance, it would seem that the P^{∞} estimates at 1.0 meter predict better than the reported 0.5 mm (-3.0 log meters) with the more ceiling-central estimates at or around 0.3 mm (-3.5 log meters). However, upon reexamination of [30], we determined that the test setup had been positioned at the ceiling *edge* and not in the better-performing center area. Performance estimates in the far corner of the surface show values of 0.45 to 0.50 mm (-3.30 to -3.35 log meters).

Figures 6 and 7 show some numerical results from the HiBall rail experiments depicted in Figure 5. The plot on the left in Figure 6 shows the "jitter" of the real system's individual position estimates (deviation from a curve fit to the data) and the plot on the right shows the estimated steady-state covariance (P^{∞}). During our initial analysis of the real system's deviation from a line fit we came to three realizations. First, we realized that the rail was sagging several millimeters as a result of its own weight and the load of the HiBall. Second, we realized that in the real system there is inherent noise (uncertainty) in both the HiBall camera models and the LED positions. Third, we realized that we were adding mechanical noise (friction, gait, etc.) as we walked the HiBall sensor along the rails. To address the first problem we computed the deviation of the real data from a *deflection curve* as described in [24]. To address the second and third problems we increased the expected measurement noise slightly. The results are shown in Figure 6. Both actual and estimated system performance are qualitatively and quantitatively consistent under the ceiling. The performance degrades similarly in both the actual and estimated data as the HiBall sensor moves out from under the ceiling LEDs. At most points where the real system lost tracking, our performance method indicated very high or infinite P^{∞} if there was no solution.

3.1.2 A Sparse HiBall™ System

To further validate the performance estimation method, we disabled every other row of LEDs in the ceiling, effectively doubling the distance between rows. We did this both in the actual system, and in our steady-state models. Here again (Figure 7) we find the performance estimates are similar to the actual system performance. While the overall "sparse" system performance remains very good with the standard deviation range increasing only slightly, the performance is less consistent with clear peaks and valleys of accuracy that map directly to the alternating enabled or disabled rows of LEDs.

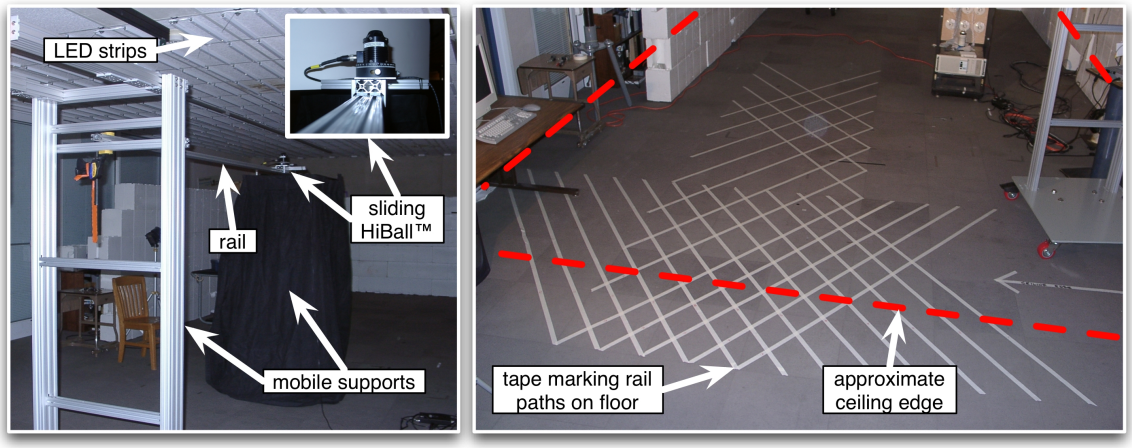

Figure 5: Experimental setup. Left: shows the LED strips on ceiling, the precision 80/20™ rail, the mobile rail supports, and the sliding HiBall™ fixture (see also the inset). Right: the tape on the floor marks the 28 different linear rail paths that collectively form a grid intended to span the edge of the ceiling (dashed red line) and some portion of the interior.

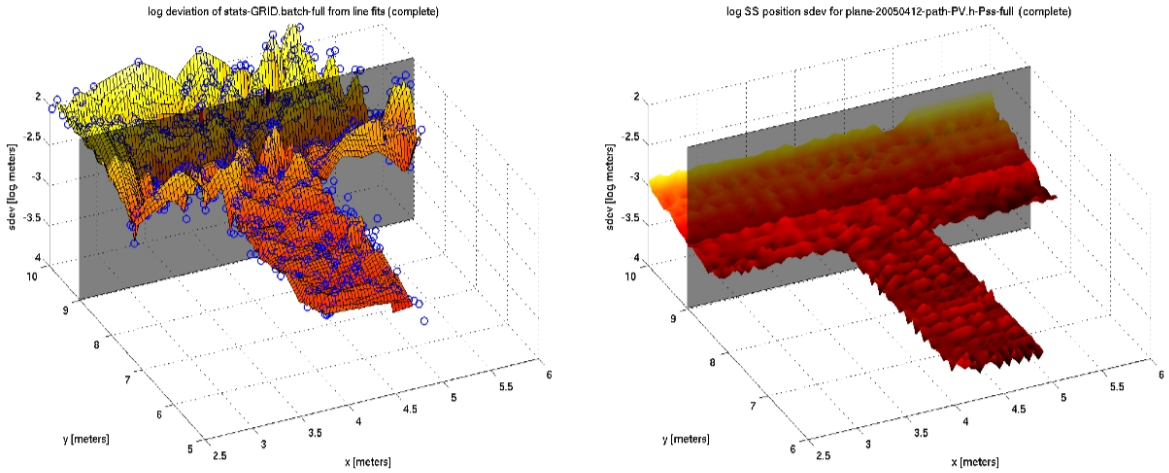

Figure 6: Comparison of actual HiBall™ position stddev with P^{∞} estimates. Left position stddev (from curve fit) of the real system. The surface is interpolated over the stddev values. The blue circles indicate actual values. Right: P_{∞} estimates over a plane fit to the rail data. The semi-transparent vertical planes in both plots marks the edge of the "ceiling" (the LED array) around 9 meters in y.

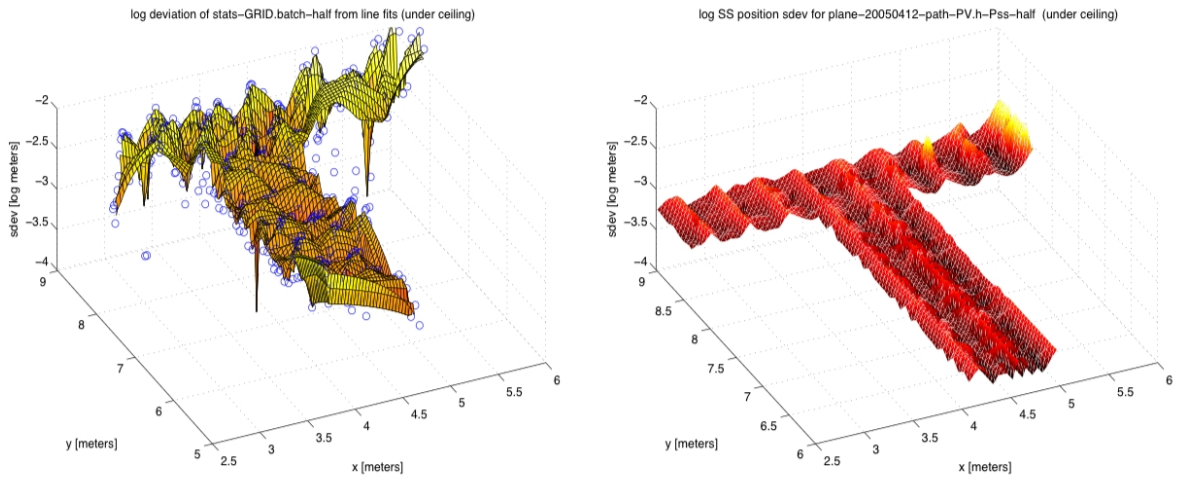

Figure 7: Comparison of actual HiBall™ position stddev (left) with P^{∞} estimates (right) when alternating rows of LEDs are disabled. See also caption for Figure 6. Note that the data off the "ceiling" was excluded from both plots to increase the dynamic range.

3.2 An Acoustic Example

Figure 1 shows the results from the performance estimation for our hypothetical acoustic system. The red spheres at the top mark the positions of the four transmitters (speakers). The system is modeled with a constant update rate of 50 Hz ($\delta t = 0.02$ seconds). The Q matrix is a 6×6 matrix of the form (12)–(14) with $qx = qy = 0.395$ and $qz = 0.107$. The measurement covariance matrix is a 4×4 diagonal matrix with $R[i,i] = 0.005\overline{z}[i]^2$ for $1 \leq i \leq 4$, where $\overline{z}$ is the range given in Section *2.1.2*. For simplicity, we chose a measurement model covariance formula that is a quadratic function of distance. We do not claim that this is the best noise model available and suggest that the measurement model can be modified to represent any function desired.

3.3 A Multi-Camera Acquisition System

We modeled an existing eight-camera acquisition rig that we use for 3D computer vision-based scene reconstruction research. (See Figure 8.) A current goal related to that research is to determine how to rearrange cameras, and we are using our tools to evaluate different configurations.

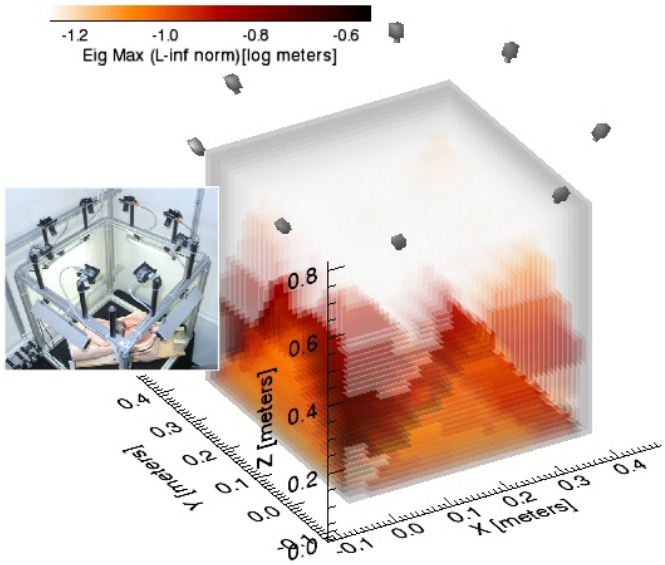

Figure 8: A volume rendering of a P^∞ analysis of an 8-camera vision-based acquisition rig. The grey camera icons st the top indicate accurate positions and rotations of the cameras.

The process model we used was the same as in the preceding acoustic example, with an update rate of 30 Hz. For the measurement function we used a simple pinhole camera model based on the Point Grey Dragonfly^{TM} camera specifications used (6 mm focal length, 640x480 resolution), where the image coordinates are given by

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \vec{x}[1]/\vec{x}[3] \\ \vec{x}[2]/\vec{x}[3] \end{bmatrix}$$

where $\vec{x} = R(\overline{x}_i - \overline{c})$, R is a camera rotation matrix, $\overline{x}_i$ is the 3D point where we are computing P^∞, and $\overline{c}$ is the camera position vector. Since our measurement function is non-linear, we used the linear approximation given by the measurement Jacobian,

$$H = \begin{bmatrix} \frac{\partial u}{\partial \overline{x}_i[1]} & \frac{\partial u}{\partial \overline{x}_i[2]} & \frac{\partial u}{\partial \overline{x}_i[3]} \\ \frac{\partial v}{\partial \overline{x}_i[1]} & \frac{\partial v}{\partial \overline{x}_i[2]} & \frac{\partial v}{\partial \overline{x}_i[3]} \end{bmatrix}$$

For R we used a function similar to that of the preceding acoustic example, where the measurement error covariance increases as the square of the distance.

In Figure 8 one can see that the view frusta of the individual cameras are recognizable, as the performance increases in the overlapping regions. In this configuration, the best performance is located at the horizontal center at a height of 0.24 meters. The worst performance below the first frusta overlap (0.68 meters) occurs at the bottom corners of the volume. Note that this figure uses a reverse color map where better performance (lower uncertainty) is represented by the brighter areas and lower performance (higher uncertainty) is represented by the darker areas. This color mapping was chosen for improved visibility.

3.4 Mixed Devices (A Hybrid System)

To illustrate the generality of our approach, we combined the acoustic model of Section 3.2 with the multi-camera system model of Section 3.3. Given the pseudo-code at the end of Section 2.3, we simply added the camera device list and parametric measurement model to the loop, along with the acoustic devices and model. The results are shown in Figure 9 (with reverse color map), where one can see how overall P^∞ decreases (performance improves) when compared to either the acoustic or multi-camera systems alone. Further, the system can now "see" at heights above 0.68 meters due to the inclusion of the acoustic sensors. In fact, the region of peak performance has moved from the horizontal center (camera) and the points closest to the speakers (acoustic) to the top four corners of the working volume (height of 0.68 meters). Cf. Figure 1.

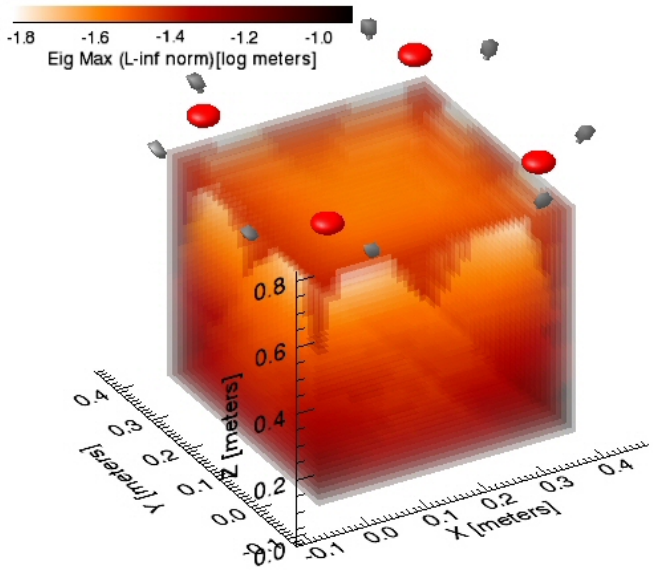

Figure 9: A P^∞ visualization for a hypothetical hybrid system with acoustic devices and cameras. Acoustic transmitters and cameras are depicted with small red and gray icons.

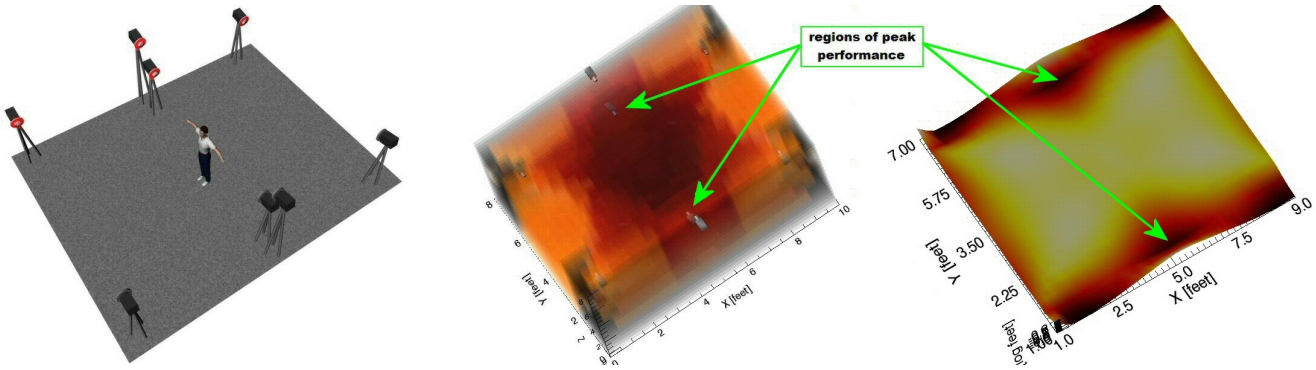

Figure 10: A hypothetical eight-camera motion capture system. Left: picture depicting the arrangement of the eight cameras in the room. Center: volume visualization of P^{∞} throughout the space. Right: Surface plot at plane of peak performance (z=5.6 feet).

3.5 A Motion Capture System

Optical systems used for human motion capture typically follow a common paradigm. Cameras are position at fixed locations in the environment, looking inward toward the capture area, so that they can observe active or passive (reflective) targets affixed to the moving human(s). As such the modeling for a P^{∞} analysis is very similar to the preceding multi-camera acquisition system.

Figure 10 depicts a hypothetical eight-camera motion capture system, and a corresponding P^{∞} visualization. While the absolute P^{∞} values may not be accurate in this analysis (and so are not included) the relative results from the volume visualization show us that, as we might expect, there are dead spots in the corners and better performance in a hexagonally-shaped pillar running through the center of the volume. Numerical analysis showed that peak performance occurs at a height of 5.6 feet, between the camera pairs at heights of 5 feet and 7 feet. A surface visualization at this height clearly shows that the areas of estimated peak performance are close to the camera pairs as indicated in Figure 10, and *not* in the center of the capture area.

4. CONCLUSIONS

Our motivation for pursuing this work originated with our own needs. In our attempts to explore variations of existing systems to improve performance and/or reduce infrastructure, we found ourselves unable to develop intuition about the effects different changes would have. When considering a design we would wonder for example, what would happen if we increased the focal length, added a camera, or combined it with accelerometers. That lack of intuition made even path-dependent simulations of alternatives very difficult. Such simulations are time consuming, and so you want to plan/choose carefully. And when you are done, such simulations often produce results that are difficult to compare, or leave you wondering. For example, you wonder what would happen if you simulated a different path, turned the target in a slightly different way, sped up the motion, etc. And while you might try and use Monte Carlo methods to explore many different paths, synthesizing appropriately realistic motion paths is difficult.

Our goal with this paper has been to share our new performance estimation approach and to convince readers that it is useful and valid. Further we have presented both the fundamental concepts and several concrete examples in the hope that readers might be able to apply the approach on their own.

We continue to improve the Artemis prototype, and to use it ourselves. While the closed-form solution is faster than iterative approaches or exhaustive simulations, one of our goals is to reduce P^{∞} computation time to allow for rapid real-time interaction with system design parameters. To this end we are investigating approaches such as parallel, adaptive, demand-driven computation to improve the responsiveness of the application. We hope to make Artemis available to others, along with a repository of process and measurement models to use as building blocks in system design and analysis.

5. ACKNOWLEDGEMENTS

We want to acknowledge Professor Gary Bishop for discussions related to this work, and Erica Stanley for her contributions to the prototype system. This research was supported by National Library of Medicine contract N01-LM-3-3514: "3D Telepresence for Medical Consultation: Extending Medical Expertise Throughout, Between and Beyond Hospitals."

6. REFERENCES

[1] B. D. Allen, G. Bishop, and G. Welch. Tracking: Beyond 15 minutes of thought: Siggraph 2001 course 11. In *Computer Graphics*, Annual Conference on Computer Graphics & Interactive Techniques. ACM Press, Addison-Wesley, Los Angeles, CA, USA (August 12-17), siggraph 2001 course pack edition, 2001.

[2] R. Ashby. *Design for an Intelligence-Amplifier*, volume 34 of *Annals of Mathematics Studies, Automata Studies*. Princeton University Press, 1956.

[3] R. Ashby. *Design For a Brain*. John Wiley & Sons, New York City, N. Y., 2nd ed. edition, 1960.

[4] D. K. Bhatnagar. Position trackers for head mounted display systems: A survey. Technical Report TR93-010, University of North Carolina at Chapel Hill, 1993.

[5] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering: with MATLAB Exercises and Solutions*. Wiley & Sons, Inc., third edition, 1996.

[6] G. Burdea and P. Coiffet. *Virtual Reality Technology*. John Wiley & Sons, Inc., first edition, 1994.

[7] X. Chen. *Design of Many-Camera Tracking Systems For Scalability and Efficient Resource Allocation*. PhD thesis, Stanford University, 2002.

[8] V. L. Chi. Noise model and performance analysis of outward-looking optical trackers using lateral effect photo

diodes. Technical Report TR95-012, University of North Carlina at Chapel Hill, April 1995.

[9] L. Davis, E. Clarkson, and J. P. Rowland. Predicting accuracy in pose estimation for marker-based tracking. In *Proceedings of Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 28–35. Institute of Electrical and Electronics Engineers, IEEE Computer Society Press, October 2003.

[10] E. Foxlin, Y. Altshuler, L. Naimark, and M. Harrington. Flighttracker: A novel optical/inertial tracker for cockpit enhanced vision. In *Proceedings of Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 212–221. Institute of Electrical and Electronics Engineers, IEEE Computer Society Press, November 2004.

[11] E. Foxlin, M. Harrington, and G. Pfeifer. Constellation: A wide-range wireless motion-tracking system for augmented reality and virtual set applications. In M. F. Cohen, editor, *Computer Graphics*, Annual Conference on Computer Graphics & Interactive Techniques, pages 371–378. ACM Press, Addison-Wesley, Orlando, FL USA, SIGGRAPH 98 conference proceedings edition, 1998.

[12] E. Fuchs/Foxlin. *Inertial Head-Tracking*. M.s. thesis, Massachusetts Institute of Technology, 1993. by Eric Foxlin, before he changed his name from Fuchs. Thesis (M.S.)–Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1993. Supervised by Nathanial I. Durlach.

[13] M. S. Grewal and P. A. Angus. *Kalman Filtering Theory and Practice*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ USA, 1993.

[14] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ USA, 2000.

[15] R. S. Kalawsky. *The Science of Virtual Reality and Virtual Environments*. Addison-Wesley Publishing Company, first edition, 1993.

[16] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[17] S. Khoury, A. Freed, and D. Wessel. Volumetric modeling of acoustic fields in CNMAT's sound spatialization theatre. In *IEEE VIS '98: Proceedings of the IEEE conference on Visualization '98*, pages 439–442. Institute of Electrical and Electronics Engineers, IEEE Computer Society Press, 1998.

[18] D. Manolakis. Efficient solution and performance analysis of 3-d position estimation by trilateration. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1239–1248, October 1996.

[19] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374. ACM Press/Addison-Wesley Publishing Co., 2000.

[20] K. Meyer, H. Applewhite, and F. Biocca. A survey of position trackers. *Presence (the Center for Research in Journalism and Mass Communication)*, 1(2):173–200, 1992.

[21] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

[22] A. Mulder. Human movement tracking technology. Technical Report TR 94-1, School of Kinesiology, Simon Fraser University, July 1994.

[23] G. Olague and R. Mohr. Optimal camera placement for accurate reconstruction. *Pattern Recognition*, 35(4):927–944, 2002.

[24] J. E. Shigley and L. D. Mitchell. *Mechanical Engineering Design*. McGraw-Hill, Inc., 1983.

[25] A. Stettner and D. P. Greenberg. Computer graphics visualization for acoustic simulation. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 195–206. Association of Computing Machinery, ACM Press, Addison-Wesley, 1989.

[26] I. E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the 1968 Fall Joint Computer Conference, AFIPS Conference Proceedings*, volume 33, part 1, pages 757–764. Thompson Books, Washington, D.C., 1968.

[27] G. Welch and G. Bishop. An introduction to the kalman filter. Technical Report TR95-041, University of North Carolina at Chapel Hill, Department of Computer Science, 1995.

[28] G. Welch and G. Bishop. SCAAT: Incremental tracking with incomplete information in places. In T. Whitted, editor, *Computer Graphics*, Annual Conference on Computer Graphics & Interactive Techniques, pages 333–344. ACM Press, Los Angeles, CA, USA (August 3-8), SIGGRAPH 97 conference proceedings edition, 1997.

[29] G. Welch and G. Bishop. An introduction to the kalman filter: SIGGRAPH 2001 course 8. In *Computer Graphics*, Annual Conference on Computer Graphics & Interactive Techniques. ACM Press, Addison-Wesley Publishing Company, Los Angeles, CA, USA, SIGGRAPH 2001 course pack edition, August 12-17 2001.

[30] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Colucci. The hiball tracker: High-performance wide-area tracking for virtual and augmented environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST)*, pages 1–11. Association of Computing Machinery, ACM Press, Addison-Wesley Publishing Company, December 1999.

[31] G. Welch and E. Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics Applications*, 22(6):24–38, 2002.

[32] H. J. Woltring. New possibilities for human motion studies by real-time light spot position measurement. *Biotelemetry*, 1:132–146, 1974.