Building Performance Visualization Using Augmented Reality

Ali M. Malkawi, Ravi S. Srinivasan
Department of Architecture, School of Design
University of Pennsylvania, Philadelphia, PA, USA
{malkawi, sravi}@design.upenn.edu

Abstract

Magnetic motion trackers have been widely used for tracking user head / hand pose information owing to their advantages such as size, occlusion-less tracking environment and high sample rate. Yet, issues such as latency and jitter leave magnetic tracker technology unfavorable as compared to other tracker technologies. While latency is due to tracker hardware employed, jitter is due to magnetic field distortion caused by the presence of metals nearby. These issues permit the emergence of registration errors when employed for Virtual Environment (VE) systems, specifically for interactive Virtual Reality (VR) or Augmented Reality (AR) applications. This paper discusses the integration of prediction-smoothing algorithms to achieve accurate registration for a magnetic tracker-based immersive Augmented Reality – Computational Fluid Dynamics (CFD) environment. In this project, Kalman and Gaussian filters are utilized to remove latency and jitter effects. In addition, to allow efficient control of head-pose data prediction, a control variable is appended to the Kalman dynamic equation. Furthermore, to permit real-time latency calibration during immersive visualization, speech-recognition is integrated with the system. Such integration, apart from enabling accurate calibration of AR setup, permits robust on-the-fly latency calibration, thereby facilitating effective user experience during interactive, immersive AR visualization of CFD datasets of indoor spaces.

Keywords: *Magnetic motion tracker, registration, augmented reality, Kalman filter, building simulation.*

1. INTRODUCTION

Virtual Environments (VE) allow users to experience their settings so as to visualize, navigate and manipulate objects in real-time using high-end visualization and interactive devices. Virtual Reality (VR) and Augmented Reality (AR) facilitate users to interact by means of complete or partial immersion, respectively. In a VR system, owing to the control of visual, and in some cases, aural and proprioceptive senses, the user experiences complete immersion. Contrastingly, AR permits the user to move in actual-space, as well as, interact with augmented graphical objects necessitating a sense of presence [1].

Virtual Environments have been widely employed for a variety of applications ranging from scientific visualization to computer games. One such application is the visualization of CFD simulation results using VEs. CFD simulations are extensively used in aerospace, nuclear, automotive, biomedical, environmental and building design-construction industries. Immersive visualization of CFD simulation allows designers to fine-tune their work based on performance results, ahead of building them in reality. Such '*immersive building simulation' by* way of integration of VEs with building data such as CFD will enhance the comprehension of simulated results [2]. Applications of VEs for CFD visualization includes the virtual wind tunnel that allowed visualization of particles as streamlines, path lines, volume arrows, etc. [3]; immersive visualization for structural analysis [4]; immersive real-time fluid simulation [5]; building performance visualization [6-8] etc. Besides, enhanced multimodal interactions for data manipulation have enabled joint-performance of tasks by humans and computers or Human-Computer Interaction (HCI) [9]. Some of the HCI modalities that enable better accessibility and effective data manipulation for VEs include speech-recognition, gesture-recognition mechanisms, wands, joysticks, six Degrees of Freedom (DoF) mouse etc. Owing to its naturalness and simplicity, speech-recognition techniques have been utilized in supporting word processing [10], CFD visualization [11] etc. However, such interactive immersive visualization using VEs rely on the immersiveness of the system employed for better comprehension of data.

Immersiveness of AR systems largely depends on accurate registration of virtual objects with the real world such that the scene matches with user perception. Accurate registration demands precise information such as the precise location of user, or otherwise, the position-orientation (6 DoF) information of the user's head. This is due to the fact that the views generated by the virtual camera should coincide with user's perception of the actual space for effective immersive experience. Therefore, any mis-registration will prevent the user from seeing the virtual and real objects as fused together, thus prompting paramount importance to (a) the selection of appropriate motion tracker technology and (b) registration inaccuracies during the motion tracking process. Motion tracker technologies include mechanical, inertial, acoustic, magnetic, optical, radio and microwave sensing. The selection of appropriate motion tracker depends on the task to be performed, and pursues several possible objectives [12], such as *sense of presence*, perceptual stability, no occurrence of simulation sickness and no degradation of task performance. Incorrect super-positions of virtual objects in actual-space are referred to as *registration errors*; they are of two types – static and dynamic. While static registration errors are due to optical distortion, system errors such as mechanical misalignments, jitter and incorrect viewing parameters, and dynamic registration are largely due to system latency and jitter [13]. In addition, these registration errors vary according to the tracker technology employed and the task to be performed.

Magnetic motion trackers have been widely used for tracking user movements in VEs due to their size, relatively higher sample rate, occlusion-less, price etc. Magnetic trackers rely on measurements of the local magnetic field vector generated by the *transmitter*, either via Direct Current (DC) or Alternate Current (AC) systems. The generation of near-field, low-frequency magnetic field vectors are sensed with 3 collocated reception antennas bound within a *sensor*. The 6DoF output from the sensor can be applied to generate new views. Despite excellent advantages, magnetic trackers have their weaknesses such as latency and jitter. Latency is due to the asynchronous nature at which the sensor measurements are tracked and their corresponding pose estimates are available to the visualization system, Figure 1 [12]. Jitter is due to the presence of ferrous or electronic devices in the surrounding and noise in the AC/DC system. Motion tracking takes a finite time-step to acquire data of user head-pose, compose and generate new sets of graphical objects based on user's viewpoint, and pose them to the Head Mounted Device (HMD) for immersive visualization. This can be disadvantageous, especially, for interactive AR applications in which the user may perceive the virtual objects lag behind, thus leading to defective task performance, as well as, motion sickness.

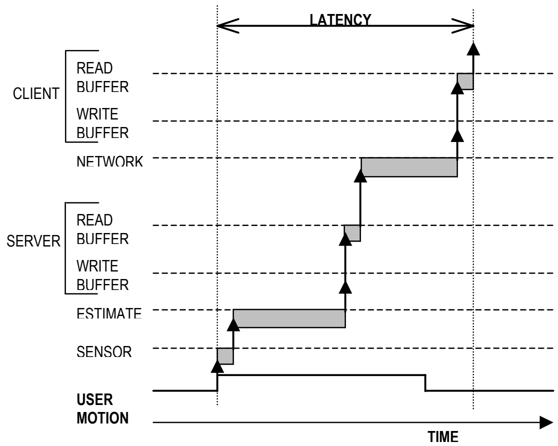

Figure 1: Tracker dataflow pipeline and latency.

In spite of vast developments in motion tracker and computer hardware technologies, precise registration of virtual objects to actual-space remains a principal concern. For magnetic trackers, this is largely due to latency and jitter.

This paper discusses the integration of prediction - smoothing algorithms to achieve accurate registration for an immersive AR-CFD environment. In this paper, a predictor-corrector algorithm – Kalman filter is integrated with an existing interactive AR system to remove latency issues. Additionally, Gaussian filter has been incorporated to remove the jitter created by the field distortion and interference by ferrous metals in the surroundings. The robust magnetic tracking system has been integrated with existing interactive, immersive AR system to facilitate manipulation of CFD datasets. To allow efficient control of head-pose data prediction, a control variable is appended to the Kalman dynamic equation. In addition, to permit real-time latency calibration during immersive visualization, speech-recognition is integrated with the system. Such integration allowed accurate registration of graphical objects in actual-space, both in static and dynamic states, and permitted effective user experience. Moreover, it allowed robust on-the-fly latency calibration during immersive building simulation.

2. ROBUST MAGNETIC MOTION TRACKING SYSTEM DYNAMICS

The robust magnetic motion tracking for immersive AR visualization utilizes two routines – Kalman filter routine and Gaussian filter routine, Figure 2. While Kalman filter routine predicts future head position-orientation estimates thus removing latency effects, Gaussian filter routine smoothens the data to eliminate the jittering effect. For robust tracking, the raw tracker data is filtered using Kalman and Gaussian routines before being sent to the AR visualization pipeline.

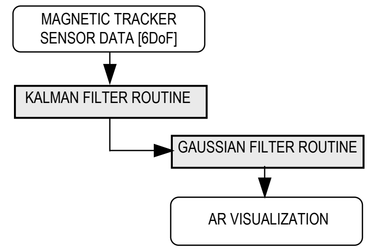

Figure 2: Robust magnetic motion tracking system.

2.1 Kalman Filter Routine

Kalman filter is a recursive solution to discrete data-linear filtering problem [14]. Kalman filter uses statistical models to weigh each new measurement relative to past information, and operates with any number of state vectors. Owing to its prediction of *a priori* estimates for future time-steps, this filter has been widely applied in military and civilian navigation systems. Earlier attempts include prediction of head pose using ten state dynamic model of HMD orientation [15]; using a modified complementary Kalman filter applied to orientation measurements [16]; prediction for inertial trackers [17] and optical trackers [18]. Kalman filter consists of (a) time update equations [predictor algorithm] for projecting forward the current state and error covariance estimates to obtain *a priori* estimates for the next time step, and (b*)* measurement update equations [corrector algorithm] for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate [19]. Kalman filter can be thought of as a predictor-corrector algorithm that avoids many of the computations and storage requirements by retaining only those values that are essential for processing future observations.

2.1.1 Kalman Filter Method Integration

In the system developed, the magnetic sensor attached to the HMD tracks the head-pose information of the user. This data is filtered by Kalman algorithm in real-time. Based on the current state estimates, Kalman filter predicts the next head-pose data, Figure 3. This enables the reduction of time lag central to the AR system, as well as, allows accurate registration of virtual objects in actual-space.

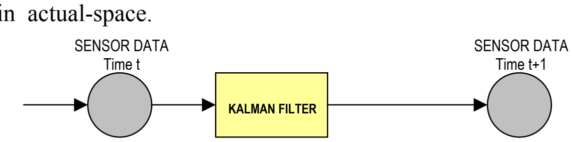

Figure 3: Kalman filter integration.

Kalman filter presents a linear, unbiased and minimum error variance recursive algorithm to optimally estimate the unknown state of a linear dynamical system from data taken at discrete time intervals. The prediction of head-pose data by Kalman filter can be considered as a two-step process. In the first step, an *a priori* new state is estimated based on previous state estimates (predictor equations). By observing the present state, an error correction is performed to the *a priori* new state estimate to optimize and generate *a posteriori* new state estimate (corrector equations), as shown in Figure 4.

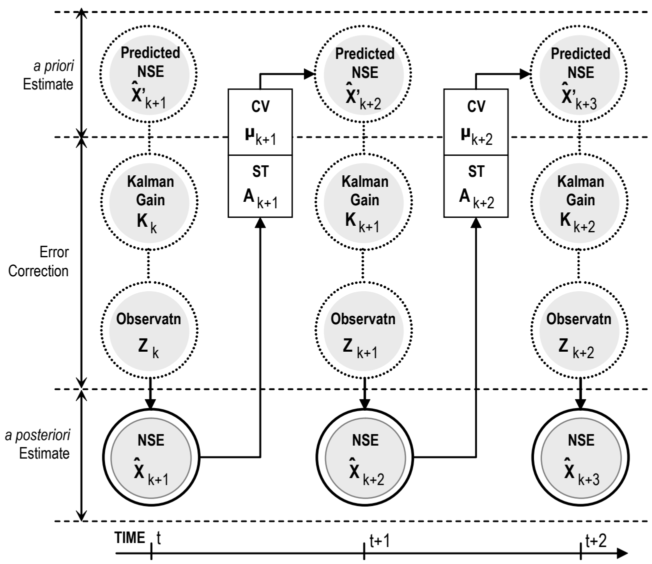

Figure 4: Kalman filter system dynamics.
[NSE – New State Estimate; CV – Control Variable; ST – State Transition]

A priori new state estimate is computed by propagating forward the current state estimate. In order to manipulate the prediction of new state estimates, a control variable is introduced to the prediction equation. In this project, the control variable is the ratio of covariance of true data and predicted data. Thus, the *a priori* predicted new state estimate is,

$$\hat{X}'_{k+1} = A_k \hat{X}_k + B_k \mu_k$$

$\hat{X}'_{k+1}$ - *A priori* future 6DoF data

$\hat{X}_k$ - Current 6DoF data

A_k , B_k - Transition matrices

μ_k - Control variable

At every discrete time interval, the error covariance is computed considering a random white noise measurement. The error covariance is utilized to update the Kalman Gain (KG). KG is directly proportional to the uncertainty in the estimate and inversely proportional to uncertainty in the measurement. KG is utilized to compute new optimal state estimates, as well as, update error estimates, Figure 5. The prior error covariance and Kalman Gain equations are,

$$P'_{k+1} = A_k P_k A_k^{\ T} + Q_k$$

$$K_{k+1} = P'_{k+1} H_k^{\ T} (H_k P'_{k+1} H_k^{\ T} + R_k)^{-1}$$

P'_{k+1} - Prior error covariance

P_k - Error covariance

Q_k - Random white noise

K_{k+1} - Kalman Gain

H_k - Matrix-valued Hessian function

R_k - Measurement error covariance

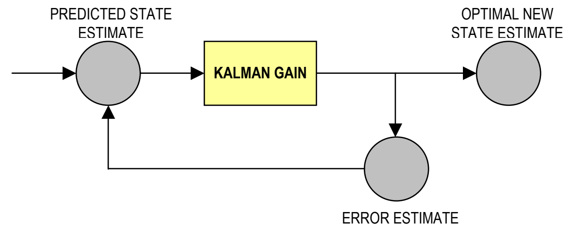

Figure 5: Kalman Gain utilization.

With the Kalman Gain, the *a priori* state estimate is updated for errors. The optimal new state estimate or *a posteriori* estimate is the summation of predicted estimate and an error term. The error term comprises of the error based on current observations. This cycle is repeated in real-time to predict future head-pose estimates. Thus, the *a posteriori* new state equation is,

$$\hat{X}_{K+1} = \hat{X}'_{k+1} + K_k (Z_k - H_k \hat{X}'_{k+1})$$

$\hat{X}_{K+1}$ - *A posteriori* future 6DoF

Z_k - Current 6DoF observation

In this routine, Kalman filter addresses the general problem of trying to estimate the state vector of a discrete-time controlled process that is governed by the linear stochastic difference equation. Applying this routine, the head pose data at time *t* is filtered through Kalman algorithm to obtain new future head-pose data. Some of the Kalman filter functions utilized for this project include *getStateVector()* to obtain state vector; *filter()* to evaluate the likelihood function for the state-space model, *update()* to update the state covariance matrix at every time step, *setQ()* to set new values to control variable in the a priori state estimate equation etc.

2.1.2 Real-time, On-line Latency Calibration

Although the integration of Kalman filter has alleviated the issue of latency, there is a need to manipulate the future head-pose prediction outputs during real-time immersive visualization. It is highly difficult for AR user wearing a HMD to access a mouse or keyboard to manipulate the control variable attached to the *a priori* state estimate equation. Changes to the control variable can be utilized to calibrate the latency inherent to tracker hardware. To access and modify the value of the control variable by the user in real-time, speech-recognition technique has been incorporated, Figure 6.

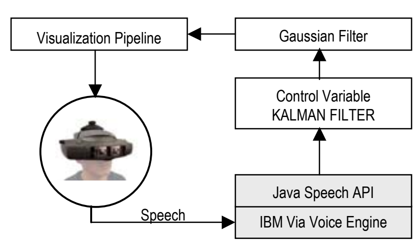

Figure 6: On-line latency calibration.

In this project, the built-in microphone with the HMD aids in transmitting user's speech to the IBM Via Voice speech-engine [20]. The corresponding values are posted to the control variable successively, in real-time using the *setQ()* method. This facilitates on-line, on-the-fly latency calibration by the AR user during immersive visualization. The predicted head position-orientation data is streamlined to the Gaussian filter routine to eliminate any undesired jitter effects. A comparison of predicted and actual head-pose data for 6DoF is presented in section 3.

2.2 Gaussian Smoothing Routine

In this routine, Gaussian filter is applied over predicted 6DoF data to remove any jitter effects caused by the possible presence of metals and or noise associated with the transmitter. Data smoothing involves the sliding of a convolution kernel over the predicted head-pose data in real-time. In this project, the 6 DoF real-time tracker data can be represented as,

$$R_t = \{t_1(x, y, z, a, b, c)\}_1^{\infty}$$

Where x_1,y_1,z_1 are sensor positions; a_1, b_1, c_1 are sensor orientations along X,Y,Z at time t=1.

A single-dimensional Gaussian filter is utilized to smoothen the tracker data, Figure 7.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-x^2}{2\sigma^2}}$$

σ – Standard Deviation
Mean of the distribution is zero [centered].

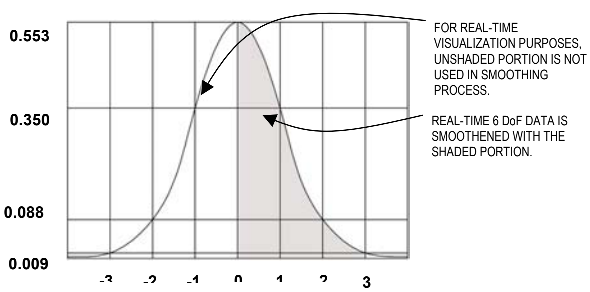

Figure 7: Gaussian filter kernel.

The convolution is performed by sliding the above kernel G(x) over R, in real-time to obtain smoothened, jitter-free calibrated tracker data. The smoothened head pose data is utilized to generate corresponding graphical views for immersive visualization.

$$R = \sum R(t - x).G(x)_{t=1}^{\infty}$$

6 DoF data { PositionX, PositionY, PositionZ, AngleX, AngleY, AngleZ}
* Gaussian Filter Kernel {0.553, 0.350, 0.088, 0.009}

3. EXPERIMENTAL RESULTS

The integration of Kalman and Gaussian filters removed the undesirable latency and jitter effects. Evaluation of the integrated system performance is vital to understand its effectiveness in predicting future head-pose estimates. The integrated system was found to be performing sufficiently well in terms of providing an uninterrupted *sense of presence* and eliminating registration errors between the virtual and real objects. In this section, the comparison of Kalman filter prediction data and actual magnetic sensor data is provided for all 6DoF head-pose data ranges, Figures 8-13.

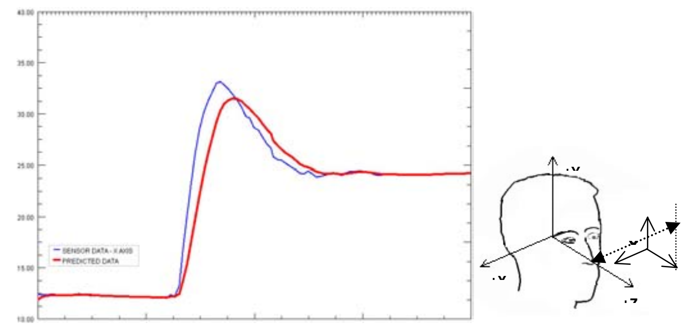

Figure 8: Sensor angle prediction (red) along X-axis.

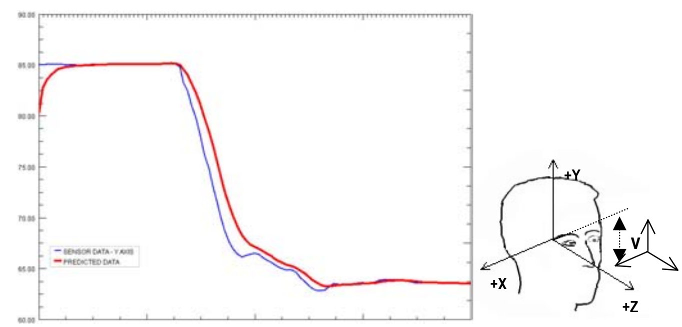

Figure 9: Sensor angle prediction (red) along Y-axis.

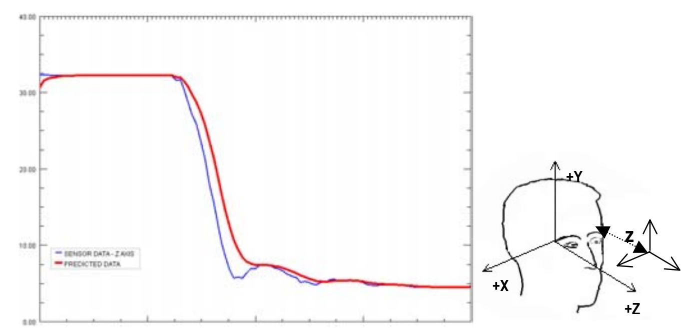

Figure 10: Sensor angle prediction (red) along Z-axis.

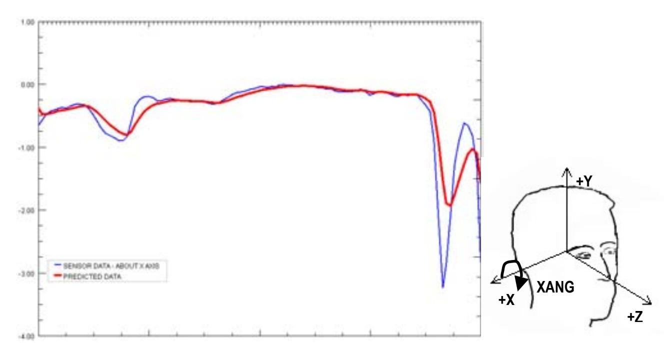

Figure 11: Sensor angle prediction (red) about X-axis.

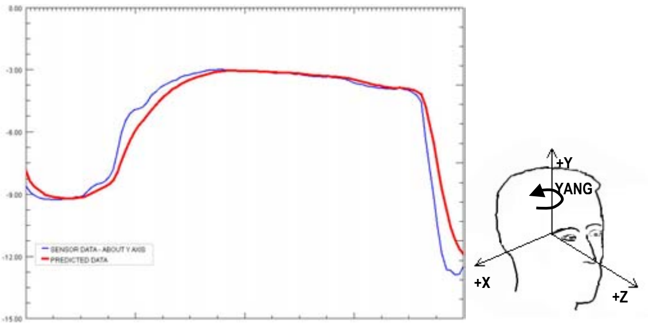

Figure 12: Sensor angle prediction (red) about Y-axis.

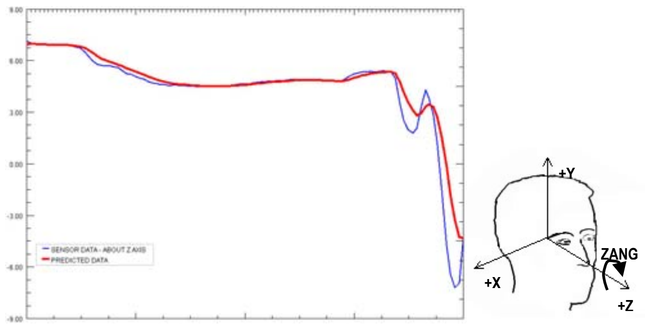

Figure 13: Sensor angle prediction (red) about Z-axis.

In addition, the integration of speech-recognition enabled effective real-time manipulation of control variable values, thereby lending efficient latency calibration during immersive visualization, Figure 14.

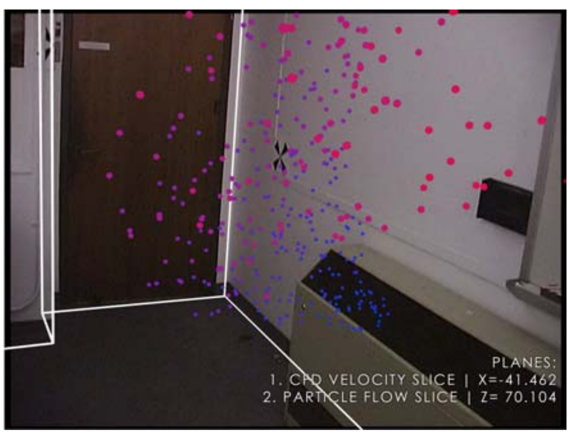

Figure 14: Registration of virtual white lines and graphical objects in actual-space, as seen from HMD.

4. INTEGRATION TO INTERACTIVE, IMMERSIVE AR VISUALIZATION SYSTEM

The prediction-smoothing algorithms were integrated to a larger AR-CFD framework to enable robust motion tracking with magnetic motion trackers for immersive building simulation [21,22]. The AR-CFD framework for building simulation encompasses three modules (1) CFD analysis module that generates CFD datasets, (2) HCI module that comprises of a library of standardized glossary of gesture recognition tasks, and (3) AR visualization module that aids in immersive visualization of CFD results in actual-space, Figure 15.

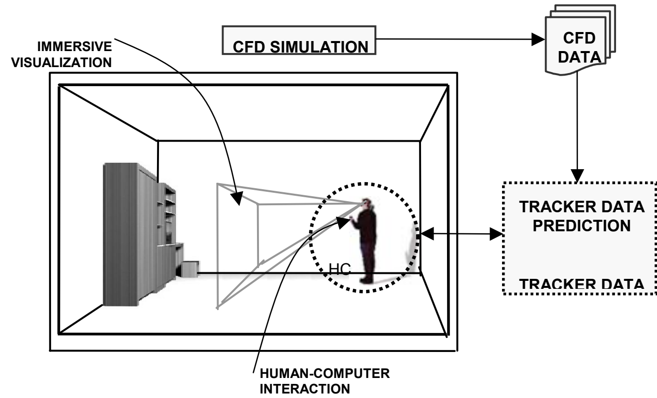

Figure 15: Immersive Building Simulation framework.

As the boundary conditions change, results of the CFD simulations are stored in Virtual Reality Modeling Language (VRML) format. These results represent corresponding iso-planes and iso-surfaces of the indoor thermal environment. Unique identifiers are attached to these VRML slices for quick display onto HMD based on user's hand gestures. Hand gestures are transformed into a set of functions that facilitate better manipulation of CFD datasets. Apart from being an intuitive approach, gestures are space-related modality that supports communication of concrete and spatial content. Some of the hand shapes developed are, 'Closed_Fist', '*Open_Flat_Palm*', '*Touch_Finger*', '*Shoot_Hand*' and '*Inc_Thumb*' that correspond to 'grab a plane', 'creating new iso-plane', 'information gathering', 'focusing on specific space/area', and 'enlarging graphical information' respectively, table 1.

Table 1: Hand shapes and corresponding functions.

Representation	Hand Shape	Function
	Closed_Fist	Grab & move an iso-planes
	Open_Flat_Palm	Create new iso-planes
	Touch_Finger	Gather information
	Shoot_Hand	Focus on specific area
	Inc_Thumb	Enlarge graphics

In this project, CyberGlove [23] is employed to transform posture data into a general description of hand-shape through forward kinematics to compute hand segment positions from given joint angles. Gestures were studied to determine the minimum number of joint angles necessary to ensure the uniqueness of the gesture. Addition of speech-recognition has rendered the AR-CFD system multimodal. Speech-recognition is utilized for real-time latency calibration. Moreover, synergistic uses of speech and gesture recognitions enhance the naturalness and facilitate better accessibility for effective data manipulation during immersive visualization.

AR visualization module assists in the visualization of CFD datasets with the support of magnetic position-orientation trackers and a HMD. *Flock of Birds* magnetic trackers [24] were employed to track the user's head pose information. These trackers have a static accuracy of 1.8mm (position-RMS) and 0.5˚ (orientation-RMS) over a verified range of 20.3cm to 76.2cm. A catadioptric (Cathode Ray Tube) CRT-based HMD [25] was utilized to visualize virtual objects in actual-space. The interactive immersive AR system maps the CFD result onto actual-space through the HMD. The user can navigate through the space, inquire about its thermal conditions, manipulate the results of data displayed, as well as, interact with the actual environment. The development of the system included specialized software written in C++, Java APIs apart from JSAPI [26] for speech recognition and JMSL [27] for Kalman filter objects.

5. CONCLUSIONS

This paper presented a method to achieve robust motion tracking with magnetic motion trackers through the integration of prediction-smoothing algorithms. The integration of Kalman and Gaussian filters to the existing AR system has improved the registration of virtual objects to actual-space, thus eliminating errors associated with latency and jitter. Moreover, the addition of real-time latency calibration technique through direct manipulation of control variable values has considerably enhanced the reliability of magnetic-based motion trackers for immersive AR applications.

Rapid user movement, both translation and rotation, allows the emergence of instantaneous errors in predicted head-pose information as compared to tracker data, as seen in Figure 11. Future study will incorporate methods that would allow the elimination of such errors. One approach is to introduce a multi-state Kalman filter that would switch between different states depending on user's head movement in 6DoF. This will provide magnetic motion trackers more reliable, accurate and tractable for most of the VE applications.

6. REFERENCES

[1]Azuma, R., 1997. A Survey of Augmented Reality. Presence: Teleoperators and Virtual Environments 6 (4): 355-385.

[2]Malkawi AM [2004]. Immersive Building Simulation [chapter]. In 'Advanced Building Simulation' (eds) Malkawi AM, Augenbroe G, Taylor Press [In press].

[3]Bryson S [1993] Virtual Wind Tunnel: A high-performance virtual reality application. In: Proceedings of IEEE Annual Virtual Reality International Symposium, pp:20-25.

[4]Impelluso T [1996] Physically Based Virtual Reality in a Distributed Environment, Computer Graphics, 30(4): 60-61.

[5]Wasfy TM and Noor AK [2001] Visualization of CFD results in immersive Virtual Environments. Advances in Engineering Software 32, pp 717-730.

[6]Shahnawaz V, Vance J and Kitti S [1999] Visualization of Post processed CFD data in a Virtual Environment. In: Proceedings of DETC99, ASME Design Engineering Technical Conference, September 1999, pp 1-7.

[7]Linden E, Hellstrom J, Cehlin M and Sandberg M [2001] Virtual Reality presentation of temperature measurements on a Diffuser for Displacement Ventilation. In: Proceedings of 4th International Conference on Indoor Air Quality and Energy Conservation in Buildings, Vol. II, pp:849-856.

[8]Malkawi AM, Srinivasan RS [2004]. A new Human-Building Interaction Paradigm: Use of Building Simulation, Augmented Reality and HCI. [Submitted to Automation in Construction Journal, Elsevier Publications. In review].

[9]Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong & Verplank, HCI. ACM SIGCHI curricula for HCI, 1996.

[10]L.R. Karl, M. Pettey, & B. Shneiderman,, Speech versus mouse commands for word processing: an empirical evaluation. International Journal of Man-machine Studies, 39, 1993, 667-687.

[11]T.M. Wasfy & A.K. Noor, Rule-based natural-language interface for VE. Advances in Engineering Software 33, 2002, 155-168.

[12]Welch G, Bishop G [2003]. An Introduction to Kalman Filter, TR 95-041, Dept of Comp Sc, University of NC at Chapel Hill.

[13]Azuma R. (2001). AR: approaches and technical challenges. In: Barfield W, Caudell T, Fundamentals of wearable computers and AR, Lawrence Erlbaum Associates Publishers, London, pp 27-66.

[14]Kalman RE [1960]. A New Approach to Linear Filtering and Prediction Problems. In: Transactions of the ASME – Journal of Basic Engineering, pp 35-45.

[15]Azuma R, Bishop G [1994]. Improving static and dynamic registration in an optical see-through HMD. In: Proceedings of 21st International SIGGRAPH, ACM, pp 197-204.

[16]Foxlin E [1996]. Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter. In: Proceedings of Virtual Reality Conference [VR 99], IEEE.

[17]Chai L, Nguyen K, Hoff B, Tyrone V [1999]. An Adaptive Estimator for Registration in Augmented Reality. In: Proceedings of Second IEEE and ACM International Workshop on Augmented Reality.

[18]Dorfmuller-Ulhaas K [2003]. Robust Optical User Motion Tracking using a Kalman Filter. In: Proceedings of VRST 2003 Conference.

[19]Welch G, Foxlin E. Motion Tracking: No Silver Bullet, but a Respectable Arsenal, In: IEEE Computer Graphics and Applications, 2002.

[20]IBM [2004] IBM Via Voice speech engine, IBM Inc. URL: http://www.ibm.com

[21]Malkawi AM, Srinivasan RS, Jackson R, Yi YK, Chan KH and Angelov S [2004]. Interactive, immersive visualization for indoor environments: Use of Augmented Reality, Human-Computer Interaction and Building Simulation. In Proceedings of Information Visualization IEEE International Conference, London. (In press).

[22]Malkawi AM and Srinivasan RS [2004]. Multimodal Human-Computer Interaction for immersive visualization: Integrating speech-gesture recognitions and Augmented Reality for indoor environments. In: Proceedings of IASTED Computer Graphics and Imaging (CGIM) Conference, Hawaii. (In press).

[23]VTI [2003] Virtual Technologies In., website accessed on August 2003. URL: http://www.immersion.com

[24]Ascension-Tech [2003]. Ascension Technologies Inc., website accessed on August 10, 2003. URL: http://www.ascension-tech.com

[25]NVIS [2003] Nvision Inc., website accessed on June 24, 2003. URL: http://www.nvisionindustries.com

[26]JSAPI [2004] Java Speech API, Sun Inc. URL: http://java.sun.com

[27]JMSL [2004] Numerical Library for Java, Visual Numerics Inc. URL: http://www.vni.com

About the authors

Ali Malkawi is a professor at the Department of Architecture, School of Design, University of Pennsylvania, Philadelphia, USA. His contact email is malkawi@design.upenn.edu

Ravi Srinivasan is a Ph.D. student at the Department of Architecture, School of Design, University of Pennsylvania, Philadelphia, USA. His contact email is sravi@design.upenn.edu