**Satoru Emura**
t00648@star.rcast.u-tokyo.ac.jp
Research Center for Advanced
Science and Technology (RCAST)
The University of Tokyo

**Susumu Tachi**
tachi@star.t.u-tokyo.ac.jp
Dept. of Mathematical Engineering
and Information Physics
Faculty of Engineering
The University of Tokyo

# Multisensor Integrated Prediction for Virtual Reality

## Abstract

Unconstrained measurement of human head motion is essential for HMDs (head-mounted displays) to be really interactive. Polhemus sensors developed for that purpose have deficiencies of critical latency and low sampling rates. Adding to this, a delay for rendering virtual scenes is inevitable. This paper proposes methods that compensate the latency and raises the effective sampling rate by integrating Polhemus and gyro sensors. The adoption of quaternion representation enables us to avoid singularity and the complicated boundary process of rotational motion. The ability of proposed methods under various rendering delays was evaluated in the respect of RMS error and our new correlational technique, which enables us to check the latency and fidelity of a magnetic tracker, and to assess the environment where the magnetic tracker is used. The real-time implementation of our simpler method on personal computers is also reported in detail.

## 1 Introduction

The synchronization of a user's motion in actual space and virtual space is significant in virtual reality (VR), because under the situation, where time lag between these spaces is greater than 100 ms, users tend to feel motion sickness and to perceive such a virtual environment as less interactive.

Motion sensors used in VR systems are classified into two groups. One is the mechanical link-type sensors like BOOM, and the other is the unconstrained sensors like Polhemus Tracker. The latter has the advantage of unconstrainedness and wider motion range, but has deficiencies of low sampling rates (~60 Hz) and critical latency (~100 ms). Although Polhemus "Fastrak" is improved in the respect of sampling rate (~120 Hz), communication delay (~40 ms with RS232C), and an ad-hoc linear prediction filter, it is still insufficient for interactiveness in the virtual environment. Most VR systems use raw output of their motion sensor in spite of the decrease in interactiveness by this latency, or try ad-hoc filtering in order to compensate for the latency such as linear extrapolation from both smoothed measurements and crude estimates of its instantaneous differential of the user's motion.

Friedmann, Starner, and Pentland (1992) pointed out that by such an ad-hoc approach, the user's quick motion causes poor prediction that overshoots the actual motion and forces users to make slow deliberate motions. They solved this problem on translational motion by means of the prediction based on the optimal linear estimation theory. However, they ignored rotational motion, which is far more critical for HMD. Liang, Shaw, and Green (1990) proposed

Presence, Vol. 7, No. 4, August 1998, 410–422

to adopt the equivalent angle-axis representation for rotational motion, but they dealt a single rotation by decomposing into four independent system models, and applied Kalman filter theory. The derivative term that appeared in their state vector was something other than angular velocity, and their prediction had no kinematic ground.

Our main idea is to combine the sensor of angular velocity to compensate for the latency of measurement of the absolute posture of a human head. In order to integrate two different types of sensors, we derived a kinematically consistent system model. The methods we proposed can not only compensate for the latency, but also raise the sampling rate through integration of multisensor information. The integration algorithm used maximum-likelihood estimation, which is also the basis of Kalman filter theory.

We proposed multisensor integration approach in Section 3. We used quaternion representation to avoid singularity and complicated boundary processing. The parameters for the proposed methods were decided based on careful inspections in Section 4. In order to check the performance of these methods in detail, we proposed the new powerful evaluation method of motion-tracking sensors based on correlation function in Section 5. The outputs of these methods with various rendering delay were computed offline and checked by RMS error and the correlational method in Section 6. In Section 7 we also reported real-time implementation of the simpler method into ordinary personal computers. The validity of this sensing system combined with a see-through HMD was checked by video recording.

## 2 Quaternion

### 2.1 Quaternion representation

We adopt quaternion (Goldstein, 1980) instead of Euler angles as the representation of orientation. The reasons are

1) when a human body rotates over $2\pi$ radian, the Euler angle exceeds the measure range of the magnetic tracker, $-\pi \sim +\pi$ and the management around the boundary is complex
2) the system model of human rotational motion described by Euler angles has singular points.

Let $v$ be a vector in 3-D space. If we rotate the coordinate by ZYX Euler angle $\gamma$, $\beta$, $\alpha$, the new description of $v$ satisfies

$$v = R_z(\gamma)R_y(\beta)R_x(\alpha)v'. \tag{1}$$

It is known that rotation is also represented by complex $2 \times 2$ matrix in general as $3 \times 3$ rotational matrix (Goldstein, 1980). This matrix consists of four real numbers $q_0 \sim q_3$.

$$Q = \begin{pmatrix} q_0 + iq_3 & q_2 + iq_1 \\ -q_2 + iq_1 & q_0 - iq_3 \end{pmatrix} \tag{2}$$

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1. \tag{3}$$

We call $q_0 \sim q_3$ quaternion. If we rotate the coordinate by ZYX Euler angle $\gamma$, $\beta$, $\alpha$, each rotation is represented by these $2 \times 2$ matrices.

$$Q_z(\gamma) \equiv \begin{pmatrix} e^{i\gamma/2} & 0 \\ 0 & e^{-i\gamma/2} \end{pmatrix} \tag{4}$$

$$Q_y(\beta) \equiv \begin{pmatrix} \cos\beta/2 & \sin\beta/2 \\ -\sin\beta/2 & \cos\beta/2 \end{pmatrix} \tag{5}$$

$$Q_x(\alpha) \equiv \begin{pmatrix} \cos\alpha/2 & i\sin\alpha/2 \\ i\sin\alpha/2 & \cos\alpha/2 \end{pmatrix}. \tag{6}$$

Consider $V$ below instead of the vector $v$. Then the rotation by Euler angle can be described in another form.

$$V = \begin{pmatrix} z & x - iy \\ x + iy & -z \end{pmatrix} \tag{7}$$

$$Q \equiv Q_x(\alpha)Q_y(\beta)Q_z(\gamma) \tag{8}$$

$$V = Q^{\dagger}V''Q \tag{9}$$

The quaternion of $Q$ is given by

$$q_0 = \cos\frac{\gamma}{2}\cos\frac{\beta}{2}\cos\frac{\alpha}{2} + \sin\frac{\gamma}{2}\sin\frac{\beta}{2}\sin\frac{\alpha}{2}, \tag{10}$$

$$q_1 = \cos\frac{\gamma}{2}\cos\frac{\beta}{2}\sin\frac{\alpha}{2} - \sin\frac{\gamma}{2}\sin\frac{\beta}{2}\cos\frac{\alpha}{2} \quad (11)$$

$$q_2 = \cos\frac{\gamma}{2}\sin\frac{\beta}{2}\cos\frac{\alpha}{2} + \sin\frac{\gamma}{2}\cos\frac{\beta}{2}\sin\frac{\alpha}{2} \quad (12)$$

$$q_3 = \sin\frac{\gamma}{2}\cos\frac{\beta}{2}\cos\frac{\alpha}{2} - \cos\frac{\gamma}{2}\sin\frac{\beta}{2}\sin\frac{\alpha}{2}. \quad (13)$$

Let $\omega = (\omega_z\omega_y\omega_x)$ rad/sec be angular velocity in body coordinate. Let $q_0 \sim q_3$ be the quaternion at time $t$, $p_0 \sim p_3$ be the quaternion at time $t + dt$, and $dQ$ be the $2 \times 2$ matrix corresponding to the rotation between $t$ and $t + dt$. $dQ$ is given in terms of $\omega$ by

$$dQ \simeq \begin{pmatrix} 1 + i\frac{\omega_z}{2}dt & \frac{\omega_y}{2}dt + i\frac{\omega_x}{2}dt \\ -\frac{\omega_y}{2}dt + i\frac{\omega_x}{2}dt & 1 - i\frac{\omega_z}{2}dt \end{pmatrix}, \quad (14)$$

and $dQ\,Q$ is given in terms of $p_0 \sim p_3$ by

$$dQ\,Q = \begin{pmatrix} p_0 + ip_3 & p_2 + ip_1 \\ -p_2 + ip_1 & p_0 - ip_3 \end{pmatrix}. \quad (15)$$

Define $F'$ as below.

$$F' \equiv \begin{pmatrix} -q_3 & -q_2 & -q_1 \\ q_2 & -q_3 & q_0 \\ -q_1 & q_0 & q_3 \\ q_0 & q_1 & -q_2 \end{pmatrix} \quad (16)$$

Combining (14) and (15), and using (16), we obtain the process dynamics expressed by quaternion and angular velocity in body coordinate.

$$(p_0\,p_1\,p_2\,p_3)^T = (q_0\,q_1\,q_2\,q_3)^T + F'\omega\frac{dt}{2} \quad (17)$$

If we adopt Euler angle representation, the process dynamics is expressed as

$$r = (\gamma\ \beta\ \alpha)^T \quad (18)$$

$$r' = r + \begin{pmatrix} \cos\alpha/\cos\beta & \sin\alpha/\cos\beta & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ \cos\alpha\tan\beta & \tan\alpha\tan\beta & 1 \end{pmatrix}\omega dt. \quad (19)$$

Apparently, this equation has singular points at

$$\beta = (\tfrac{1}{2} + n)\pi. \quad (20)$$

The displacement of Euler angle becomes infinite though the angular velocity is finite. We can easily avoid this singularity problem by the quaternion representation.

## 3 Multisensor Integration

Prediction is often used for compensation of delay. Kalman filter theory is well known as the method of designing an optimal prediction filter taking into account the complex motions of target system and measurement error. This has been applied to the delay of the magnetic tracker (Friedmann, Starner, & Pentland, 1992; Liang, Shaw, & Green, 1990). Prediction of physical quantity needs some kind of its derivative in state vector. What was measured in this research was only position or quaternion; its derivative was not measured. We expected the improvement of estimates by adding the measurement of those derivatives.

This demands the kinematically consistent system model. As for rotational motion, the relation between Euler angle or quaternion and angular velocity is needed. For that purpose, we derived the approximated linear system model, which describes the relation between quaternion and angular velocity explicitly.

Kalman filters have only one measurement model, which means that the filter enables the integration of simultaneous measurements of various sensors, but that the estimate rate is bound to the slowest sensor at most if we integrate sensors of various sampling rates by Kalman filter. In order to improve the estimate rate, the information of the quick sensor should be reflected to the estimates rationally in some way.

In our problem, the slowest sensor is the magnetic tracker, and the quick sensor is the gyro sensor. When the measurements of both quaternion and angular velocity are available, we propose the optimal prediction by Kalman filter. As for how to reflect measurements of gyro sensor to the estimates, we first considered the

maximum-likelihood estimation of the state vector based on the measurement of angular velocity. We assumed that the system model and process noise were well known. This is the first method we propose. This method gives us the optimal prediction; however, it was expected that the computation load of maximum-likelihood estimation is a bit heavy for the target computer of real-time implementation. So, we also considered the extrapolation by integrating the displacement of quaternion calculated from the measured angular velocity. This is the second method we propose.

Let $H$ be the measurement matrix corresponding to simultaneous measurements of both the Polhemus and the gyro sensor, and let $H^g$ be the measurement matrix corresponding to the measurement of the gyro sensor only.

$w$ = process noise
$Q$ = covariance matrix of $w$
$H^p$ = measurement matrix of both Polhemus and gyro
$v^p$ = measurement noise by $H^p$
$R^p$ = covariance matrix of $v^p$
$H^g$ = measurement matrix of gyro
$v^g$ = measurement noise of $H^g$
$R^g$ = covariance matrix of $v^g$

$$F \equiv \left(\begin{array}{c|c} I & dtF' \\ \hline 0 & I \end{array}\right) \tag{21}$$

$$x_{n+1} = Fx_n + w_n \tag{22}$$

$$\overline{w_m w_n^T} = Q\delta_{mn} \tag{23}$$

$$H \equiv \left(\begin{array}{c|c} I & -TF' \\ \hline 0 & I \end{array}\right) \tag{24}$$

$$H^g \equiv (0|I) \tag{25}$$

$$y_n^p = H^p x_n + v_n^p \tag{26}$$

$$y_n^g = H^g x_n + v_n^g \tag{27}$$

$$\overline{v_m^p v_n^{pT}} = R^p \delta_{mn} \tag{28}$$

$$\overline{v_m^g v_n^{gT}} = R^g \delta_{mn} \tag{29}$$

### 3.1 Algorithm of Method I

When both the Polhemus and gyro sensor are available, the measurement model is given by equation (26). When only the gyro sensor is available, the measurement model is given by equation (27). The algorithm (Emura & Tachi, 1993) consists of two procedures. Procedure 1 is executed when both the Polhemus and gyro measurements are available. Procedure 2 is executed when only the gyro measurement is available.

#### Procedure 1

$$x_{n+1/n} = Fx_{n/n} \tag{30}$$

$$P_{n+1/n} = FP_{n/n}F^T + GQG^T \tag{31}$$

$$x_{n/n} = x_{n/n-1} + K_n^p[y_n^p - H^p x_{n/n-1}] \tag{32}$$

$$P_{n/n} = P_{n/n-1} - K_n^p H^p P_{n/n-1} \tag{33}$$

$$K_n^p = P_{n/n-1}H^{pT}[H^p P_{n/n-1}H^{pT} + R^p]^{-1} \tag{34}$$

#### Procedure 2

$$x_{n+1/n} = Fx_{n/n} \tag{35}$$

$$P_{n+1/n} = FP_{n/n}F^T + GQG^T \tag{36}$$

$$x_{n/n} = x_{n/n-1} + K_n^g[y_n^g - H^g x_{n/n-1}] \tag{37}$$

$$P_{n/n} = P_{n/n-1} - K_n^g H^g P_{n/n-1} \tag{38}$$

$$K_n^g = P_{n/n-1}H^{gT}[H^g P_{n/n-1}H^{gT} + R^g]^{-1} \tag{39}$$

The algorithm used in procedures 1 and 2 is the same as that of a discrete Kalman filter (Gelb, 1974).

### 3.2 Algorithm of Method II

The algorithm is divided into two procedures. Procedure 1 is for the extrapolation of the state vector based on the measurement of angular velocity only. Procedure 2 is for the optimal prediction based on the simultaneous measurement of both quaternion and angular velocity. Note that the sampling rate of quaternion of Method II is half of that of Method I.

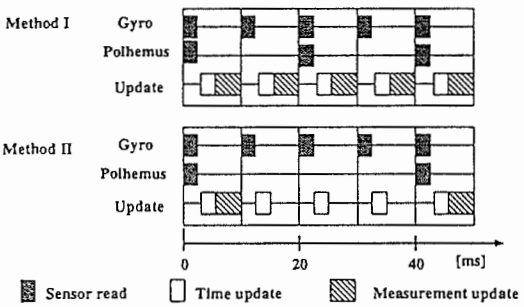


**Figure 1.** *Timing chart of available sensors and procedures in the proposed method.*

### Procedure 1

Substitute angular velocity in state vector $x_n$ of mean angular velocity within previous 30 ms

$$\overline{\omega} = (\overline{\omega_z}\ \overline{\omega_y}\ \overline{\omega_x}) \tag{40}$$

and update as below.

$$x_{n+1/n+1} = \left(\frac{r_{n/n} + \delta t \cdot F'\overline{\omega}}{\overline{\omega}}\right) \tag{41}$$

$$P_{n+1/n+1} = FP_{n/n}F^T + GQG^T \tag{42}$$

### Procedure 2

$$x_{n+1/n} = Fx_{n/n} \tag{43}$$

$$P_{n+1/n} = FP_{n/n}F^T + GQG^T \tag{44}$$

$$x_{n/n} = x_{n/n-1} + K_n[y_n - H^p x_{n/n-1}] \tag{45}$$

$$P_{n/n} = p_{n/n-1} - K_n H^p P_{n/n-1} \tag{46}$$

$$K_n = P_{n/n-1}H^{pT}[H^p P_{n/n-1}H^{pT} + R^p]^{-1} \tag{47}$$

Procedure 2 is the same as the algorithm of a discrete Kalman filter (Gelb, 1974).

Figure 1 is the timing chart of available sensors and procedures. We measured the orientation at 25 Hz and the angular velocity at 100 Hz. The output was obtained at 100 Hz.

## 4 Approximation of Parameter

### 4.1 Rendering Delay

Not only the delay of sensors but the delay by rendering must be considered in our HMD problem. Assume that we got the measurement of gyro at $t = 0$. Using the definitions below, we see the relation of observed value and estimated value. Note that the measured value of X is expressed as $X^*$.

$t_{pol}$ = delay of magnetic tracker
$t_{render}$ = delay by rendering
$(q_0\ \omega_0)$ = state vector at $t = -t_{pol}$
$(q_1\ \omega_1)$ = state vector at $t = 0$
$(q_2\ \omega_2)$ = state vector at $t_{render}$

| | $t = -t_{pol}$ | $t = 0$ | $t = t_{render}$ |
|---|---|---|---|
| Real | $(q_0\ \omega_0)^T$ | $(q_1\ \omega_1)^T$ ⇓ | $(q_2\ \omega_2)^T$ ⇓ |
| Observed | — | $q_0^*\ \omega_1^*$ | — |
| Estimated | — | $q_2^*\ \omega_2^*(=\omega_1^*)$ | — |

Before rendering the virtual scene, $q_2^*$ and $\omega_2^*$ must be known and are derived as below by using $dq$ and $d\omega$, which are statistical variables.

$$T \equiv t_{pol} + t_{render} \tag{48}$$

$$q_2 = q_0 + TF'\omega_1 + dq \tag{49}$$

$$q_2^* = q_0^* + TF'\omega_1^* \tag{50}$$

$$\omega_2 = \omega_1 + d\omega \tag{51}$$

In the state vector $(q\ \omega)^T$, there is no derivative term of angular velocity $\omega$, so the prediction like equation (50) is impossible. Instead, we decided to approximate $\omega_2^*$ by $\omega_1^*$ and to consider that the measurement error becomes a bit larger, because the output of the gyro sensor became noisy (Figure 2), and the matching between real motion and the measurement of the gyro is not so affected by the slight shift of approximation $\omega_2^*$ by $\omega_1^*$.

### 4.2 Process Noise

Covariance matrices of process noise and measurement noise were given by $Q$, $R$, and $R_G$. We assume process noise and measurement noise to be independent.

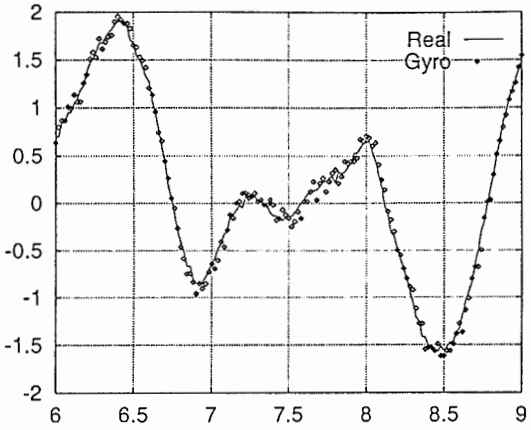


**Figure 2.** *Angular velocity of human head and its measurement by gyro sensor. The real motion "calculated" was calculated from the Euler angle measured by link-type sensor.*

Let $I_3$, $I_4$ be $3 \times 3$ and $4 \times 4$ unit matrices.

$$Q = \left(\begin{array}{c|c} w_1^2 I_4 & 0 \\ \hline 0 & w_2^2 I_3 \end{array}\right) \tag{52}$$

$$R = \left(\begin{array}{c|c} v_1^2 I_4 & 0 \\ \hline 0 & v_2^2 I_3 \end{array}\right) \tag{53}$$

$$R_G = (v_2^2 I_3) \tag{54}$$

$w_1$, $w_2$ are determined as the standard deviation of remnant error when the output of the high-precision link-type sensor (Tachi, Arai, & Maeda, 1990) sampled at 50 Hz was fitted to system model equation (30).

$$w_1 = 0.0212 \quad w_2 = 0.10[\text{rad/s}]$$

### 4.3 Measurement Noise

The measurement error cannot be determined as easily as process noise, because we must take into account the Polhemus and rendering delay. The covariance matrices we should consider are $\overline{(q_2^* - q_2)(q_2^* - q_2)^T}$ and $\overline{(\omega_2^* - \omega_2)(\omega_2^* - \omega_2)^T}$. Assume $q_0^*$, $dq$, $\omega_1^* - \omega_1$ is statistically independent.

$$\begin{aligned} &\overline{(q_2^* - q_2)(q_2^* - q_2)^T} \\ &\quad = \overline{(q_0^* - q_0)(q_0^* - q_0)^T} + \overline{dq\, dq^T} \\ &\qquad + T^2 \overline{F'(\omega_1^* - \omega_1)(\omega_1^* - \omega_1)^T F'^T} \end{aligned} \tag{55}$$

From equation (3), we approximate

$$F' F'^T \approx I. \tag{56}$$

$\overline{(q_0^* - q_0)(q_0^* - q_0)^T}$ was given as the measurement error of raw output of magnetic tracker. $\overline{dq\, dq^T}$ and $\overline{(\omega_1^* - \omega_1)(\omega_1^* - \omega_1)^T}$ was approximated by random walk.

$$\overline{dq\, dq^T} \equiv \frac{t_{pol} + t_{render}}{dt} w_1^2 I_4 \tag{57}$$

$$\overline{F'(\omega_1^* - \omega_1)(\omega_1^* - \omega_1)^T F'^T} \approx \left(\frac{T}{dt} w_2^2 + v_2^2\right) I_4 \tag{58}$$

$$\overline{(q_0^* - q_0)(q_0^* - q_0)^T} \equiv 0.0276^2 I_4 \tag{59}$$

Finally, we obtain $v_1 \sim 0.0508$ if $t_{render} = 0$.

As for $v_2$, we applied the same technique. $\overline{d\omega\, d\omega^T}$ can be approximated by random walk, and $\overline{(\omega_1^* - \omega_1)(\omega_1^* - \omega_1)^T}$ was given as the measurement error of raw output of gyro sensor.

$$\begin{aligned} v_2^2 I_3 &= \overline{(\omega_2^* - \omega_2)(\omega_2^* - \omega_2)^T} \\ &\sim \overline{(\omega_1^* - \omega_2)(\omega_1^* - \omega_2)^T} \\ &= \overline{(\omega_1^* - \omega_1)(\omega_1^* - \omega_1)^T} + \overline{d\omega\, d\omega^T} \end{aligned} \tag{60}$$

$$\overline{d\omega\, d\omega^T} = \frac{t_{render}}{dt} w_2^2 I_3 \tag{61}$$

$$\overline{(\omega_1^* - \omega_1)(\omega_1^* - \omega_1)^T} = 0.12^2 I_3 \tag{62}$$

## 5 Off-line Experiment

### 5.1 Apparatus

We used Polhemus Tracker as the sensor of orientation, and three compact gyro sensors from Murata Co. (GYROSTAR ENC-05S) as the sensor of angular velocity in the body coordinate fixed to the human head.

In order to evaluate performance, we compared the output of the proposed method with measurement of the link-type motion tracker (Tachi, Arai, & Maeda, 1990; Oyama, Tachi, & Inoue, 1993). Most linkages in this motion tracker are made not of metallic parts but FRP (fiber-reinforced plastics). The precision of the link-type sensor (accuracy 0.0125°, resolution 0.025°) was

much better than that of Tracker (accuracy 0.1°, resolution 0.5°) because the rotational angle of each joint was measured by a high-precision optical encoder and up-down counter.

The output of the up-down counter was directly connected to the data bus of the computer, and lead-time of this up-down counter was only 2 μs. CPU performance determined the delay of this system. This system with 80286 (10 MHz) measured the operator's motion and controlled the robot at least 300 Hz. Therefore, the response delay and communication delay of this link-type sensor was negligible compared with the Polhemus Tracker (~80 ms).

We checked three points:

(1) We set the source and the receiver of magnetic motion trackers as closely as possible in several runs for gathering data. The distance between the source and the receiver was always within 0.5 m.
(2) We made a testbed with a single axis for evaluating magnetic trackers. Most of the testbed is plastic, and the least-metallic parts like a potentiometer and bearings are used. We took normalized cross-correlation on this device, and compared it with the normalized cross-correlation taken on the six-DOF mechanical link system. The result is that the peak value with the six-DOF motion tracker is almost the same as the peak value with the testbed.
(3) The noise by the backlight of the HMD's LCD HMD can cause bigger skews when the distance between the receiver and the LCD is within 15 cm than the link-type motion tracker.

We conclude that the skew by the six-DOF link tracker is small enough from the above observations, and used the output as a standard signal.

### 5.2 Result ($t_{render}$ = 0 ms)

In order to make performance comparison easier to understand, we converted the estimated quaternion into ZYX Euler angle γ, β, α and compared various outputs with the standard signal (output of the link-type sensor). Figures 3 and 4 show the standard signal, the raw output of Polhemus and the output of the proposed method. The average delay of Tracker was nearly 80 ms. It is apparent that the proposed method compensated this delay well.

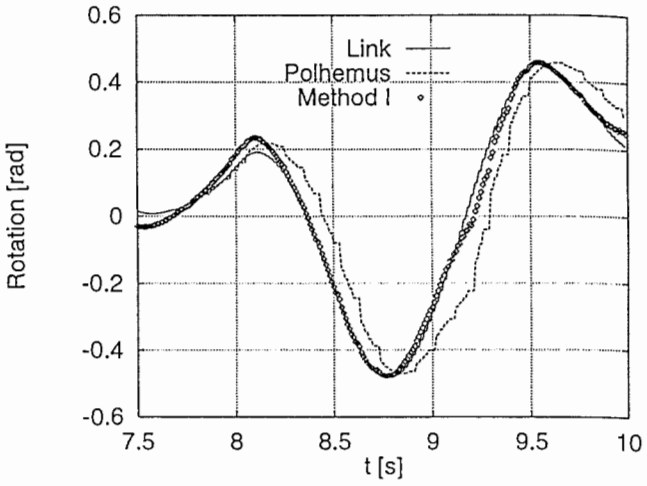


**Figure 3.** *Comparison of link data, Polhemus, and Method I (proposed).*

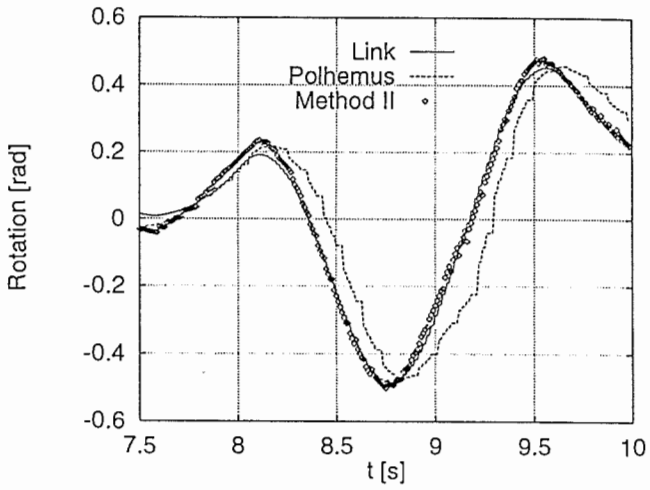


**Figure 4.** *Comparison of link data, Polhemus, and Method II (proposed).*

Table 1 shows the RMS (Root Mean Square) error of various outputs from the standard signal. The RMS error of the Kalman-filtered raw Polhemus Tracker was greater than that of the raw Tracker because of over-

**Table 1.** *Performance Comparison by RMS Error. "Polhemus" Denotes Raw Data of Polhemus Tracker, "Kalman" Denotes Kalman-filtered Raw Polhemus Data, and "Proposed" Denotes the Output of the Proposed Method*

| | RMS ($\times 10^{-2}$ [rad]) | | |
|---|---|---|---|
| | $\gamma$ | $\beta$ | $\alpha$ |
| Polhemus | 7.12 | 9.22 | 9.00 |
| Method I | 6.95 | 4.50 | 4.19 |
| Method II | 7.10 | 4.52 | 4.32 |

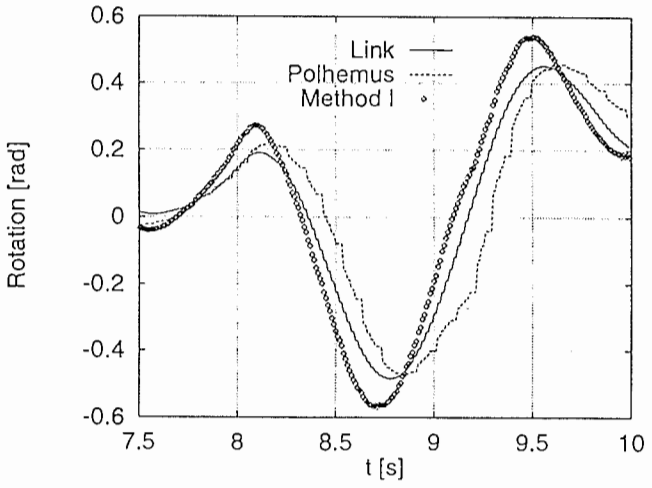


**Figure 5.** *Comparison of link data, Polhemus, and Method I (proposed).*

shoots of its prediction. RMS error of the proposed method was reduced to nearly half that of the raw Tracker output except roll angle. It was guessed that the arrangement of the metallic parts of the link-type sensor effected $\gamma$ intensively.

### 5.3 Result ($t_{render}$ = 90 ms)

We set $t_{render}$ 90 ms because the average frame rate of the virtual scene of our VR system is 11 Hz. Remark that the standard signal for RMS comparison is the output of link-type sensor shifted to $t_{render}$ left on the graph.

Figures 5 and 6 show the output of Method I and II.

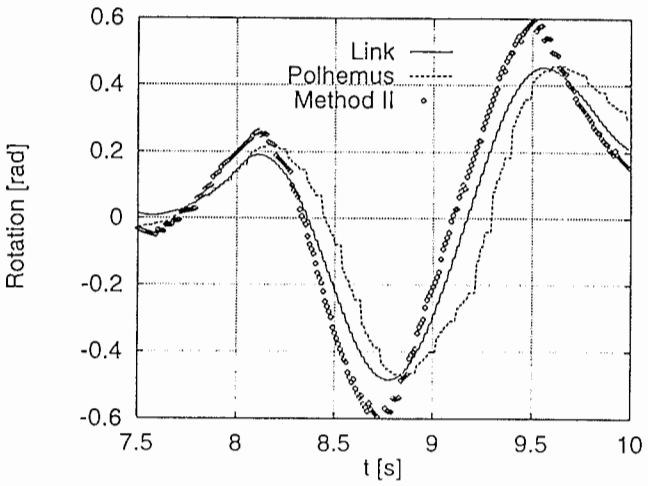


**Figure 6.** *Comparison of link data, Polhemus, and Method II (proposed).*

The output of Method I predicts 170 ms ahead of the original signal. There are the slight overshoots around the peak. The output of Method II is not always so continuous as Method I, and the overshoots around the peak are a bit greater than Method I. We see that the RMS error decreased to one-half to two-thirds of that of the original signal, and that Method I gave better estimation than Method II.

## 6 Correlational Evaluation

### 6.1 Principle

Let x(t) be the original signal, and let y(t) be the measurement of $x(t)$ by a target sensor. Cross-correlation function $\Phi_{xy}(\tau)$ (Beauchamp, 1973) is defined as

$$\Phi_{xy}(\tau) \equiv \overline{x(t)y(t+\tau)} \tag{63}$$

$$\equiv \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x(t)y(t+\tau)dt. \tag{64}$$

The normalized cross-correlation function $\rho_{xy}(\tau)$ (Beauchamp, 1973) is defined by using $\Phi_{xy}(\tau)$ as

$$\rho_{xy}(\tau) \equiv \frac{\Phi_{xy}(\tau)}{\sqrt{\Phi_{xx}(0)}\sqrt{\Phi_{yy}(0)}}. \tag{65}$$

**Table 2.** *Performance Comparison by RMS Error. Rendering Delay (90 ms) Is Considered*

| | RMS ($\times 10^{-2}$ [rad]) | | |
|---|---|---|---|
| | $\gamma$ | $\beta$ | $\alpha$ |
| Polhemus | 7.28 | 17.2 | 18.3 |
| Method I | 8.21 | 6.10 | 12.5 |
| Method II | 8.80 | 8.79 | 14.5 |

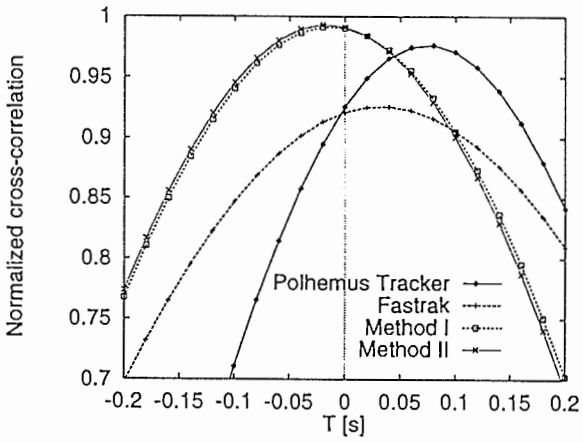


**Figure 7.** $\rho(\tau)$ *of Polhemus, Fastrak, Method I, and Method II.*

This normalized cross-correlation function has properties. First, the range of $\rho_{xy}(\tau)$ is between $-1$ and 1.

$$-1 \leq \rho_{xy}(\tau) \leq 1$$

Second, if signal $y(t)$ follows $x(t)$ with lag of $T$s,

$$x(t) \simeq y(t + T). \tag{66}$$

$\rho_{xy}(\tau)$ has a peak at $\tau \sim T$. Third, suppose independent random noise is added on signal $y(t)$. Let $S$ be the signal power, and $N$ be the noise power. Then

$$\Phi_{xx}(0) = S \tag{67}$$

$$\Phi_{yy}(0) = S + N \tag{68}$$

$$\Phi_{xy}(T) = S. \tag{69}$$

The peak of $\rho_{xy}(\tau)$ can be described

$$\rho_{xy}(T) = \frac{S}{\sqrt{\Phi_{xx}(0)}\sqrt{\Phi_{yy}(0)}} \tag{70}$$

$$= \sqrt{\frac{S}{S+N}} \tag{71}$$

$$\frac{N}{S} = \left(\frac{1}{\rho_{xy}(\tau)}\right)^2 - 1. \tag{72}$$

$\rho_{xy}(T)$ is the function of S/N ratio and can be the index of the fidelity of $y(t)$ to $x(t)$.

Therefore, we can evaluate the delay and fidelity of the target sensor quantitatively by checking the peak of its normalized cross-correlation function (Emura & Tachi, 1994).

### 6.2 Result ($t_{render}$ = 0 ms)

Figure 7 shows the $\rho(\tau)$ of Polhemus Tracker, Fastrak, and the proposed methods. We used the output of mechanical link-type sensor as the standard signal. The delay of Fastrak was half that of the Polhemus Tracker, but there remained still 30 ms of delay. There remained no delay for the proposed method, and its fidelity index was slightly improved.

### 6.3 Result (various $t_{render}$)

By the correlational technique above, we can plot the total compensated delay and the S/N ratio of the output as a function of assumed rendering delay. Note that the standard signal is not the output of the link-type sensor, but the raw output of the magnetic tracker. Figure 8 shows that Method I can compensate for a longer delay than Method II. The upper limit of compensation is 200 ms for Method I and 120 ms for Method II. If we apply Method II with $t_{render}$ = 90 ms, there still remained 50 ms delay, but it is half of the critical delay 100 ms. The N/S ratio is nearly 5% (Figure 9), and small enough for practical use.

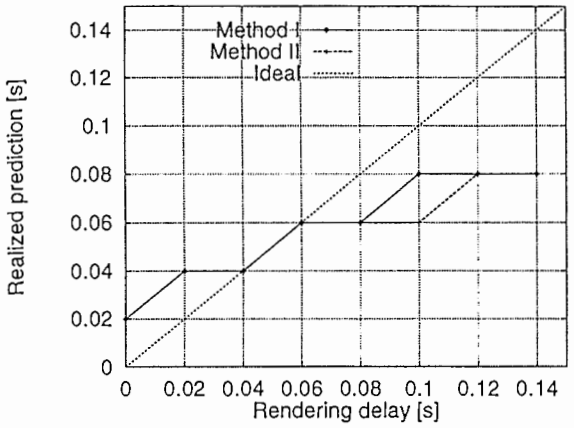


**Figure 8.** *Assumed rendering delay versus total compensation.*

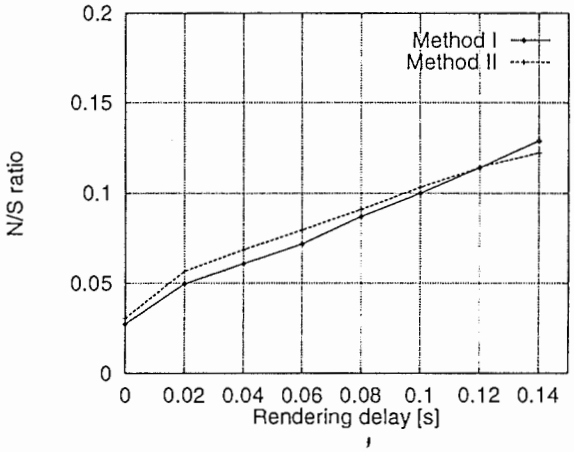


**Figure 9.** *Assumed rendering delay versus N/S ratio.*

## 7 Real-Time Implementation

### 7.1 Hardware

The multisensor system consisted of a receiver of Polhemus Tracker and three compact gyro sensors (Piezo-electric vibrating gyroscope GYROSTAR ENC-05S from Murata MFG Co.). This sensing system was attached to a see-through HMD. A PC (Intel 80486 Overdrive 40 MHz) read the measurement of Polhemus through RS232C (19.2 Kbps), and that of the gyro sensors via an AD board. See Figures 10 and 11.

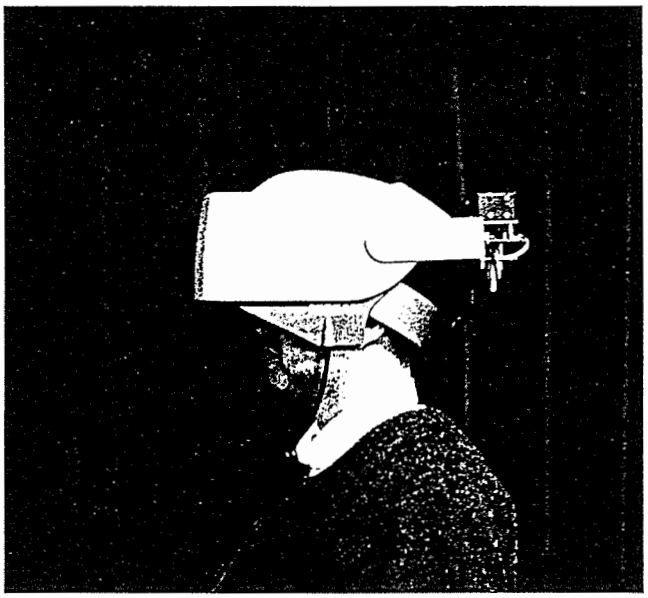

**Figure 10.** *See-Through HMD equipped with Polhemus and three compact gyros (in the box mounted at the rear of HMD).*

### 7.2 Software

In order to use the computational ability of the PC as efficiently as possible, we designed the interruption-based program that can assign I/O waiting time to other tasks. The program consisted of three tasks below.

Task A—Handling of Polhemus Tracker via RS232C
Task B—Extrapolation based on the measurement of gyro sensors (Procedure 1) Transmission of estimates via parallel printer port
Task C—Optimal prediction based on the measurement of Polhemus and gyro sensors (Procedure 2)

We divided update calculation into two tasks, because the load of the calculation of the Kalman filter was expected to be heavy, and the unit-sampling interval was critical for the PC to finish that calculation. See Figure 12.

Task A was activated by RS232C interruption. Task B was activated every 10 ms by interruption of DMA channel provided by the AD board. The measurement of the gyro was transferred by DMA channel. Task C, running in the background, requested present state vector to task B when the data from Polhemus was available, received it, calculated the algorithm of Kalman filter, and sent the

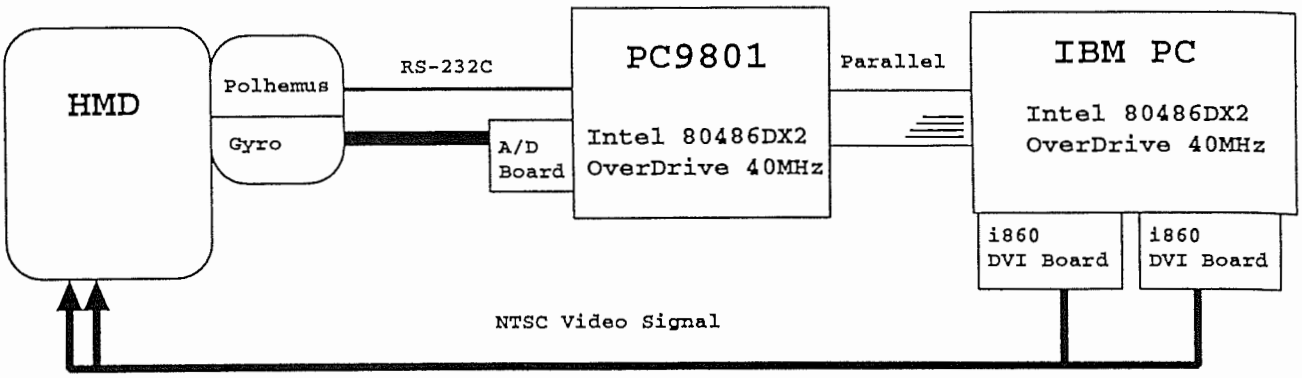


**Figure 11.** *Configuration of the real-time system.*

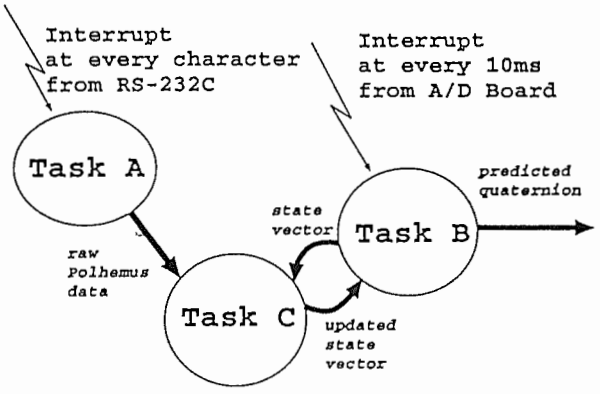


**Figure 12.** *Software Architecture.*

result to task B. Task B copied the result of task C if it was available; otherwise it kept extrapolation based on the measurement of gyro.

Three-dimensional rendering of virtual space was done on an IBM PC by World Tool Kit (Sense8 Co.). The IBM PC sent a request to PC every image frame (10 ~ 12 frames/sec.). If requested, task B sent the latest estimate of human head motion to the IBM PC via parallel port for printer.

### 7.3 Verification by See-through HMD

We made a virtual copy of an actual room (consisted of 49 polygons) and 3-D images of this room were updated 11 frames/sec. on average and communication between the PC9801 and IBM PC via parallel port took only 1.4 ms.

Syncronization between the virtual and actual room was checked by see-through HMD and video camera. We set a 8 mm video camera in the see-through HMD and recorded the real scene overlaid by the virtual scene from the position of a human eye under various rotational motions at 30 frame/sec. Then we played back the video tape frame by frame, and checked how many frames passed since the virtual scene started to follow the real motion.

In the case of the raw output of the Polhemus Tracker, the delay was 166 ~ 200 ms (5 ~ 6 frames). Our real-time method decreased the delay to 33 ~ 66 ms (1 ~ 2 frames). This fit well with the result of the off-line computation of Method II.

## 8 Discussion

There are a lot of factors in VR systems that would cause time lag such as dynamic response of the motion sensor and communication between the sensor and computer, computation for the generation of virtual spaces, and communication between computers.

Even if we could develop a sensor without any measurement delay, this would not solve all these problems. We discuss the relation between our method and these problems in this section. We also discuss the application of our method to teleoperation.

### 8.1 Multiple Tracking

Though the maximum sampling rate of Fastrak is 120 Hz, which is higher than that of our real-time implementation now, the sampling rate of the proposed methods has no upper limit in principle, and is determined only by the computation speed of computer used.

When Fastrak is tracking motions of $n$ ($1 \sim 4$) points, each sampling rate decreases to $1/n$ uniformly. However, these $n$ points are each on a part of the human body, and the sampling rates should be varied according to the speed of each part. Our method enabled us to raise the sampling rate decreased to $1/n$.

### 8.2 Generation of Virtual Scene

The real-time generation of 3-D images is one of the biggest problems in VR. It is obvious that 3-D images, built up by complicated objects or processed by computationally heavy techniques like texture mapping, give us better reality. However, the more complicated the virtual space is, the more it takes to render 3-D images and the slower the frame rate is. For us to feel interactiveness, the frame rate must be at least 10 frame/sec, where the maximum rendering delay can be estimated easily at 100 ms.

Even if we could measure human motions with 0 $\mu s$ delay, there would remain this rendering delay. Some kind of prediction is inevitable. Therefore our approach, better prediction by multisensor-integration method, will remain important if the performance of the magnetic tracker improves drastically and the measurement delay is almost 0 $\mu s$.

### 8.3 Networked Virtual Environment

In order to join people over distance through virtual environments, the sites should be networked computers. It takes some time for changes in virtual space to propagate and to synchronize on the network. In each computer, synchronization is realized by interruption processes such as network arbitration, network control, and low-level system activity, which causes latency. This should be compensated for by any kind of prediction. Though it is expected that the effects of such events will appear as time lags in nonstationary manner, the fixed time-lag compensation proposed in this paper could be the first-order approximation of the final solution to that problem.

### 8.4 Application for Teleoperation

In the situation where we connect an operator not to virtual space in a computer but to a robot in actual space ("advanced teleoperation" or "tele-presence"), the cycle time of human motion sensing needs to be less than a few milliseconds to control the slave robot smoothly. This rate is much higher than that of the Polhemus. In order to apply low-sampling-rate, remotely sensing device like the magnetic tracker, our multisensor integration approach provides a significant advantage.

## 9 Conclusions

We propose the concept of multisensor integration to solve the time-lag problem of HMDs in single virtual space. Its validity has been shown through off-line computation. The simpler algorithm was able to be implemented successfully in real-time VR system, and the performance was checked. We also proposed a correlational method for evaluating the delay and fidelity of motion-tracking sensors, and showed the relation between the fidelity and the N/S ratio of target sensors, which is very useful to assess the experimental environment where a magnetic tracker is used.

## References

Beauchamp, K. G. (1973). *Signal processing.* London: George Allen and Unwin.

Emura, S., & Tachi, S. (1993). Sensor Fusion based Measurement of Human Head Motion. *Journal of the Robotics Society of Japan,* 11(6), 161–164.

Emura, S., & Tachi, S. (1994). Sensor Fusion based Measurement of Human Head Motion. *Proceedings of the 3rd IEEE*

*International Workshop on Robot and Human Communication,* 124–129.

Emura, S., & Tachi, S. (1994). Compensation of Time Lag between Actual and Virtual Spaces by Multi-Sensor Integration. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems,* 463–469.

Friedmann, M., Starner, T., & Pentland, A. (1992). Synchronization in Virtual Realities. *Presence: Teleoperators and Virtual Environments, 1*(1), 39–144.

Gelb, A. (Ed.) (1974). *Applied Optimal Estimation.* Cambridge, MA: MIT Press.

Goldstein, H. (1980). *Classical Mechanics* (2nd ed.). Reading, MA: Addison-Wesley.

Liang, J., Shaw, M., & Green, M. (1990). On Temporal-Spatial Realism in the Virtual Reality Environment. *UIST '91,* 19–25.

Oyama, E., Tachi, S., & Inoue, Y. (1993). Experimental Study on Remote Manipulation Using Virtual Reality. *Presence: Teleoperators and Virtual Environments,* 2(2), 112–124.

Tachi, S., Arai, H., & Maeda, T. (1990). Tele-Existence Master Slave System For Remote Manipulation, *IROS '90,* 343–348.