The Visual Computer

On latency compensation and its effects on head-motion trajectories in virtual environments

Jiann-Rong Wu, Ming Ouhyoung

Communication & Multimedia Lab., Department of Computer Science and Information Engineering, National Taiwan University, Roosevelt Rd, Taipei, Taiwan, ROC 106
e-mail: ming@csie.ntu.edu.tw, chriswu@ms.digimax.com.tw

We present experiments on latency and its compensation methods in a virtual reality application using a head-motion display (HMD) and a 3D-head tracker. Our purpose is to compare, both in simulation and in a real task, four tracker prediction methods: the grey system theory-based prediction, Kalman filtering, a simple linear extrapolation, and the basic method without prediction. Typical motion trajectories of the four methods in simulation are plotted, and jitter effects are examined. Kalman filtering was found to have the largest jitter among the four. An experiment is also presented to simulate a real-world application: following a tour guide in a walkthrough of a building. An improvement of 120% was observed.

Key words: Motion prediction – Latency in HMD – Virtual reality technology

Correspondence to: J.-R. Wu

1 Introduction

In virtual reality applications, one of the major goals is to provide a computer-generated, immersive environment that is realistic with respect to appearance, behavior, and interaction. However, the requirements of such an ultimate display are not easily met [18]. One of the critical problems is the perceived latency [12], or lag. This is the time delay from the user's input action until the response becomes available for display. The time lag continues to be a major limitation in the virtual environment. In a head-mounted display (HMD) system, if the latency is relatively large (over 100 ms for instance) it will cause dizziness with long time wear for some people. At the same time, the illusion of a virtual world is destroyed if the objects on the screen jitter significantly while the head is not in motion – the "swimming effect" mentioned by [4].

The overall latency stems from the time lags present in individual components of the virtual environment (VE) systems [19]. Major components of end-to-end latency include (1) the time for internal processing by input sensors, (2) data transfer from the sensors to the host computer, (3) update of the physical simulation output by applications in response to user input, and (4) image rendering and display. To reduce the latency, through system hardware and software reorganization, Jacoby et al. [11] improve temporal response significantly by minimizing the round-trip delay. However, for a commercial HMD system, displaying from an "old" LCD screen (e.g., VR4 HMD), there is still about 170 ms of constant refreshing delay, even though the system round-trip delay has been fully eliminated. Moreover, to become more and more realistic in visualization, the complexity of the rendered environment will increase dramatically, and the latency will increase accordingly. Therefore, an efficient latency compensation method is still needed for real applications.

To compensate for latency, many proposed methods use prediction in tracking. Several HMD systems have been implemented with head-tracker prediction [1, 5, 14, 15, 17], where a "look-ahead" algorithm is implemented. It uses the 3D position and orientation as the input data. In this paper, we focus on the comparison between two prediction methods in head tracking: the well-known Kalman filtering method [15] and the grey system based method, which Wu and Ouhyoung propose [20]. A simple linear extrapolation method and the basic method without prediction were also included in our formal experiment.

The Visual Computer (2000) 16:79–90

Liang and colleages [15] developed a head-tracking prediction algorithm based on Kalman filtering. They found that the latency felt by a user is mainly due to the delay in orientation data. Based on this observation, a predictive Kalman filtering algorithm was designed to compensate for delay in orientation data. Azuma and Bishop [1] used Kalman filtering with inertial sensors mounted on a see-through HMD to improve the dynamic registration, that is, to reduce the latency. The result can significantly aid the head-motion prediction in real cases; on the average, prediction with inertial sensors produces two to three times less errors than that of prediction without inertial sensors, and five to ten times less than that of using no prediction at all. Emura and Tachi and Foxlin [8, 9] also improved tracking performance with Kalman filtering based on sensor fusion from various sources. Mazuryk and Gervautz [16] used a combination of two-step Kalman prediction and image deflection for head tracking and achieved good results. The testbeds of such improving methods all focused on small latency and got very good results. However, at bigger latency (more than 130 ms), the prediction accuracy, with or without the use of inertial sensor in Kalman filtering, is not significantly different [1]. In general applications, the overall latency should not be always constrained, and in most cases latency can be more than 130 ms (caused by low refresh rates in LCDs and high-complexity rendering environments). In this case, we chose not to use the inertia-based Kalman filtering in our experiments.

We proposed a prediction method [6, 20] using the grey system theory. Grey system theory is applied to the prediction of tracker motion of an HMD system because the behavior of the tracker output is “grey”. (*Grey* is defined as partially known). In our experience, prediction with the grey system produces 5 to 12 times less error, on the average, than prediction without the grey system. Moreover, the grey-system prediction method is computationally less complex than Kalman filtering.

Both these algorithms give reasonably good results in the tracker prediction. However, the remaining question is, if one wants to implement a virtual environment with tracker prediction, which one is more suitable and under what conditions? According to our observation, the resolution of an HMD, the system latency, and head motion (slow or jerky) in completing a task will potentially affect the results of comparison. Therefore, we want to control some of the factors in a formal experiment and try to give an answer to the question. In this paper, we compare these two methods in quantitative analysis through motion trajectory plots in the simulation, and then describe a 3D target-tracking experiment involving head motion to determine which one is more acceptable in real cases.

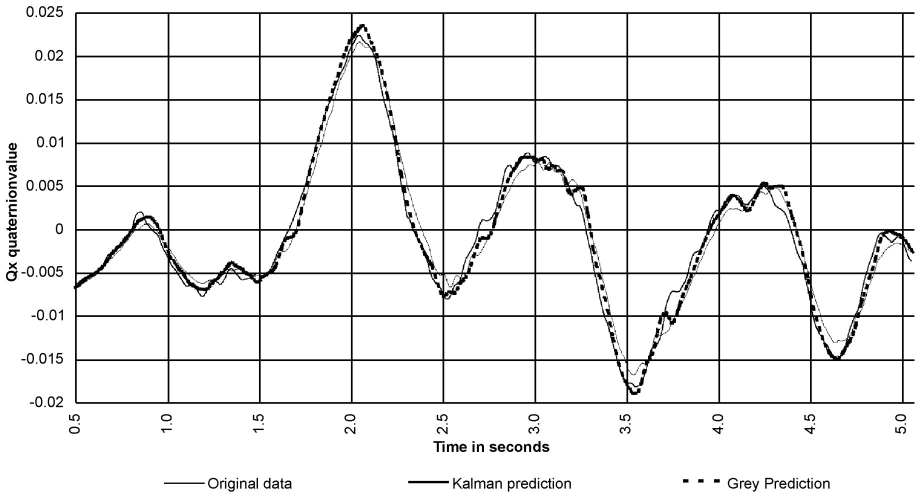
Traditionally, to deal with a digital system involving noise, digital filters are usually included to analyze the system. Prediction-based systems use signals from a 3D tracker as input, process them, and output the predicted data. For detailed descriptions of the characteristics of the grey-system method, please refer to the Appendix. For detailed descriptions of Kalman filtering-based prediction, please refer to the original papers [1, 3, 10, 13, 15, 16].

2 Quantitative analysis

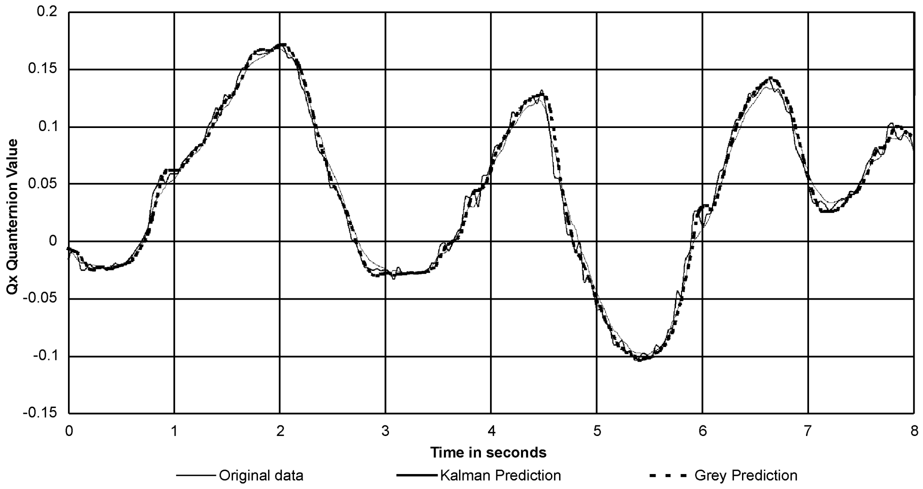
In this section, the focus is on the quantitative analysis of two prediction methods: Kalman filtering and the grey system in 3D orientation. Note that there is no difference between translation and orientation in the case of the grey system-based prediction method, but the Kalman filtering-based method does have explicit assumptions about orientation. Most cases of the behavior of these two prediction methods have similar results. Thus, we present two cases here as samples for our quantitative analysis.

Case 1. The original and the inertial sensor-based Kalman prediction data are retrieved from the recorded data of [1], included in the CD-ROM version of the SIGGRAPH’94 Proceedings. The data are from a small segment of a quaternion curve. As a high-performance optoelectronic tracking system, the system update rate is about 60 times/s, and the prediction latency is set to 60 ms. Similarly, the recorded original data is applied to the grey system; the results are shown in Fig. 1.

Case 2. The original data are also a small segment (about an eight-second head motion) of a quaternion curve, which was recorded from our electron magnetic tracker system. We apply both the Kalman filtering and the grey system to the data to get the predicted sequence. In this case, the system update rate is 25 times/s, i.e., every 40 ms, and the prediction is set to 120 ms away. A comparison of the three sets of data is shown in Fig. 2.



1



2

Fig. 1. Plot of Qx curve (quaternion) with inertia-based Kalman prediction versus grey system prediction

Fig. 2. Plot of Qx curve (quaternion) with Kalman prediction without inertial sensor versus grey system prediction

2.1 Mean square error

Table 1 shows the mean square error in the Qx curve (quaternion) with Kalman prediction (not inertia-based) versus the grey system. From the data for case 1 in Table 1, we see that the Kalman filtering with inertial sensors is a little bit better than the grey system prediction, where a prediction length of 60 ms is used, but in case 2, where a prediction length of 120 ms is used, the grey system is better than the Kalman filter.

2.2 Measurement of the jitter effect

In addition to the error in the Qx curve, the jitter effect can be measured with the fast Fourier transform (FFT). We can treat the higher-frequency noise as jitter caused by a noisy tracker system. The energy of the higher-frequency noise can be calcu-

Table 1. A table of mean square error in the Qx curve (quaternion) with Kalman prediction versus the grey system

	Kalman prediction	Grey system prediction
Case 1 (prediction at 60 ms)	0.001641 (with inertial sensor)	0.001774
Case 2 (prediction at 120 ms)	0.007955 (without inertial sensor)	0.006380

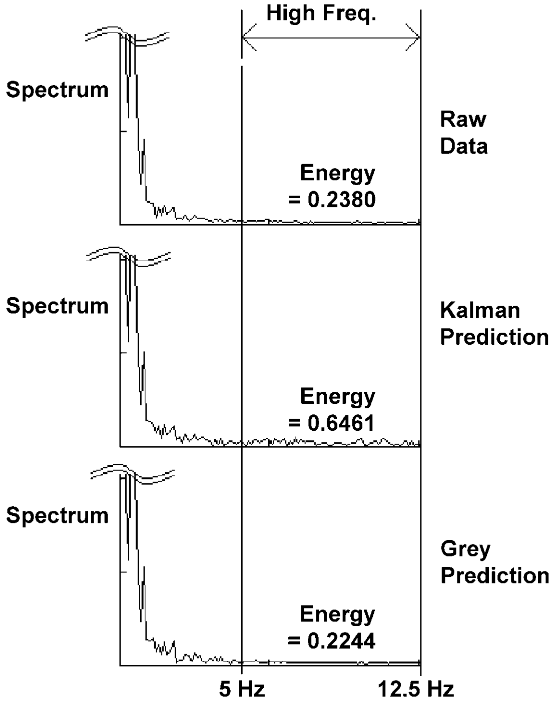


Fig. 3. The spectrums using the fast Fourier transform in case 2 and the corresponding accumulated energy covering 5 Hz to 12.5 Hz

lated to measure the intensity of the jitter. We assume that the more the jitter in the predicted trajectory, the larger the energy will be. From our observation, the U-turnlike motion of the head cannot exceed 5 times/s, 5 Hz, due to head-motion constraints. Therefore, we define "jitter" components as those higher than 5 Hz. In case 1 with a look-ahead of 60 ms, the three sequences measured have almost the same spectrum; thus, the jitter effects are similar. However, in case 2 with a look-ahead of 120 ms, the jitter effect of the Kalman-based prediction is so big that its energy is about 3 times larger than that of either the grey-system prediction or the raw data (Fig. 3). That is, at the prediction length of 120 ms, the jitter in the Kalman filtering appears to be the largest.

3 Experimental design

3.1 Experimental set-up

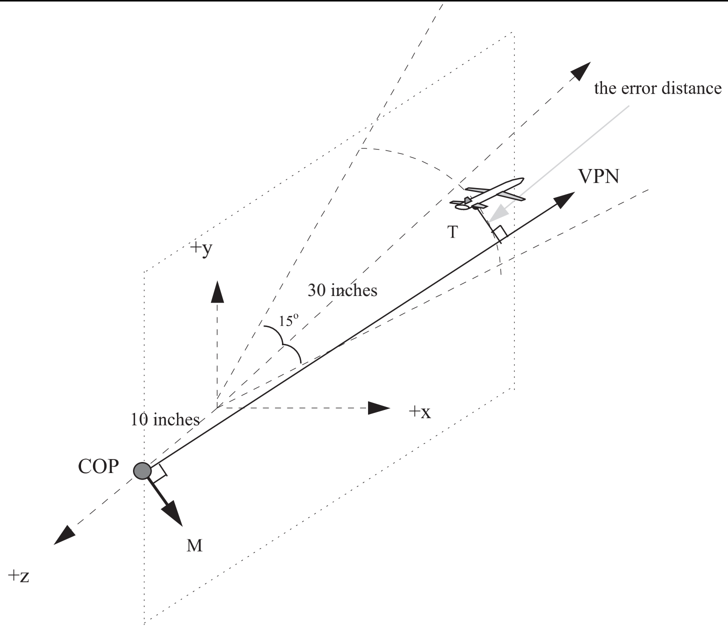
With a low-resolution HMD such as the old VPL Eyephone, there is no way to see the jitter effects, since the LCD screen is slow in response time. Because of this and the low resolution of our HMD (less than 320 × 240 pixels), we decided to use CrystalEye stereo glasses instead of the VPL Eyephone in our experimental set-up. The stereo glasses have at least half the resolution (half a scan line for each eye in stereo mode) of the SGI Indigo^2 Extreme's screen. Therefore, participants can clearly see the jitter effects. We put a Flock-of-Birds 3D tracker (Ascension, VT) on top of a headphone, and let a participant wear the stereo glasses to see in stereo. This set-up is very different from a true immersive HMD environment, but if the head rotation is limited to 15 degrees, the system functions the same way as a HMD.

In our experiment, using the SGI Indigo^2 Extreme for rendering (a teapot model consisting of 604 triangles in stereo) and a Flock-of-Birds 3D tracker, the system latency is 120 ms $\sim$ 160 ms. This is more than the 60 ms reported in [1] when they used Pixel-Planes 5. Since the prediction accuracy at 130 ms, with or without the use of the inertial sensor, in Kalman filtering is not significantly different and we are constrained in our equipment, we have chosen not to use the inertia-based Kalman filtering in the next experiment.

In our experiment, the subject is instructed to imagine participating in a dogfight simulation while wearing stereo glasses (Fig. 4). Figure 5 is a diagram of the 3D target-tracing task, where VPN is the view plane normal; COP, the center of projection; and M, the vector between two eyes. The error distance in the tracing task is defined as the distance from the center of the target T to the plane passing through COP with a surface normal M.

The Kalman filtering and the grey system used a prediction length of 120 ms with six historical samples, while the simple extrapolation method based

4



5

Fig. 4. A photo of our experimental system

Fig. 5. Diagram of the 3D target tracing-task. An error distance in the tracing task is defined as the distance from the center of the target *T* to the plane passing through View Reference Point (VRP) with a surface normal M. VPN, view plane normal; COP, center of projection; M, the vector between two eyes

on the LaGrange formula only predicted one sample distance away (40 ms).

3.2 Experiment

We are interested in the performance of subjects using the following four methods in a 3D flying target-tracing task with head-controlled orientation:

1. Nonprediction, here called ***N***.
2. Prediction with simple extrapolation based on the LaGrange method, here called ***E***.
3. Prediction with the grey system theory, here called ***G***.
4. Prediction with the Kalman filtering method, but with no inertial sensor proposed. This method, which was first proposed in 1991, is called ***K*** here.

3.2.1 Hypothesis

1. Performance of prediction methods **G** and **K** is better than that of method **N**.
2. **K** and **G** are equally good in terms of this tracking experiment, since their motion trajectory in simulation data appears to be very similar.

Two weak hypotheses, which are not considered very important in this paper, state that: (1) among the three prediction methods, **K** and **G**, having a higher order of complexity, are better than **E** and (2) **E** is better than **N**.

3.2.2 Procedure

For each trial, the subjects were told to trace the flying target as closely as possible by minimizing the distance between the center of the viewport and the center of the target in 30 s. After a short message, pressing a specific key signaled the start of each trial. A beep signaled the end of the trial. Each subject participated in two sessions, where slow and fast motions of the target were presented. The training took about 5 min for each subject, and the experiment took about 30 min. Each subject was trained with each method until additional training gave no significant improvement, usually within 5 min.

The computer automatically recorded the subject's performance in terms of error distance. Each trial was followed by a thirty-second rest period.

3.2.3 Subjects

The eight volunteer subjects were graduate students in the Computer Science Department of National Taiwan University. All subjects were able to see in stereo with the stereo glasses.

3.2.4 Design

The within-subject design is based on repeated measures, and both the multiple comparison procedures and Student-t test are used at $\alpha = 0.05$ significance level [2].

3.3 Results

Table 2 shows the error distance between the center of the viewport and the center of the flying target, which is set at 101.6 cm (40 inches) away in "slow" motion, which means ± 0.2 degree of random rotation in the y-axis every 0.04 s. Table 3 shows the error distance in "fast" motion, which means ± 0.4 degree every 0.04 s.

The multiple comparison procedures, such as Multiple F, Tukey, and Scheffé tests, are statistical tests used to find how means differ from one another in a repeated measurement design. These procedures obtain a critical value (CV), which specifies the minimal difference between two treatment means that is statistically significant at the α level chosen. The actual difference between the means in each comparison of interest is then compared with the CV.

In the ***N*** versus ***G*** test, ***N*** has a mean error of 3.827, and ***G***, a mean error of 3.142 (Table 2). Since differences between the two means is 0.68, which is greater than the critical values of Multiple F, Tukey, and Scheffé tests at $\alpha = 0.05$ (Table 4), the results corroborate with our hypothesis 1, that is, ***G*** is better than ***N*** in performance. Similarly, according to Tables 4 and 5, ***K*** is better than ***N*** both in slow and fast motion (hypothesis 1), and ***G*** is better than ***E***. ***K*** is better than ***E*** in fast motion (weak hypothesis 1 after hypothesis 1 and 2). However, weak hypothesis 2 cannot be corroborated by these

Table 2. Performance data (average error distance) in centimeters for ***N***, ***E***, ***G***, and ***K*** of a target-tracing task (slow motion)

Subject	***N***	***E***	***G***	***K***
1	8.735	7.612	6.850	7.323
2	11.748	10.851	8.941	9.108
3	9.053	8.781	8.664	7.765
4	9.911	9.136	8.011	8.796
5	10.676	9.223	8.324	8.329
6	7.684	7.501	6.543	5.974
7	9.909	8.672	8.136	8.227
8	10.046	9.688	8.385	8.077
Mean	9.720	8.933	7.981	7.950

Table 3. Performance data in centimeters for ***N***, ***E***, ***G***, and ***K*** of a target-tracing task (fast motion)

Subject	***N***	***E***	***G***	***K***
1	12.520	12.654	11.153	11.008
2	14.039	13.594	12.837	12.845
3	13.797	12.672	10.919	10.716
4	14.435	14.110	13.282	13.383
5	14.389	13.251	11.415	11.887
6	12.865	11.684	10.643	10.119
7	15.197	14.508	12.667	12.212
8	14.049	13.515	12.827	12.266
Mean	13.912	13.249	11.968	11.805

Table 4. A table of mean differences between any pair of the methods ***N***, ***E***, ***G***, and ***K*** in slow motion. The symbols a,b,c mean that the difference between two means is greater than the critical value set by (**a**) multiple F test, (**b**) Tukey test and (**c**) Scheffé test

	N	***E***	***G***	***K***
N	-	0.31	$0.68^{a,b,c}$	$0.70^{a,b,c}$
E		-	0.37	0.39
G			-	0.01
K				-

[a] $P < 0.05$, Multiple F test; critical value = 0.425
[b] $P < 0.05$, Tukey test; critical value = 0.568
[c] $P < 0.05$, Scheffé test; critical value = 0.619

Table 5. A table of mean differences between any pair of the methods ***N***, ***E***, ***G***, and ***K*** in fast motion. The symbol denotation is the same as that of Table 4

	N	***E***	***G***	***K***
N	-	0.26	$0.77^{a,b,c}$	$0.83^{a,b,c}$
E		-	0.50^{a}	$0.57^{a,b}$
G			-	0.06
K				-

[a] $P < 0.05$, Multiple F test; critical value = 0.399
[b] $P < 0.05$, Tukey test; critical value = 0.533
[c] $P < 0.05$, Scheffé test; critical value = 0.581

tests; therefore, we can not claim that $\boldsymbol{E}$ is better than $\boldsymbol{N}$.

According to the multiple comparison tests, $\boldsymbol{G}$ and $\boldsymbol{K}$ are not significantly different; thus, the results can also corroborate our null hypothesis 2, that is, the hypothesis $\boldsymbol{K} = \boldsymbol{G}$ cannot be rejected; so $\boldsymbol{G}$ and $\boldsymbol{K}$ are equally good.

3.4 Discussion

We used complete counterbalancing among participants to reduce the multiple-treatment effects, such as practice effects and treatment-carryover effects. That is, we changed the order of tests for each participant during the experiment.

1. Participants got tired after 30 min of the experiment. The symptoms: stiff neck and sore eyes reported by two participants.
2. Four of the participants showed rapid improvement during the training period in the first 5 min. Then, after 20 min, the average tracking distance error increased for the same task, which is a clear indication of fatigue. The causes of the fatigue could be (a) intensive playing and high focusing in the target-tracing task, (b) jittering objects on the screen, and (c) lack of experience with wearing shutter glasses for stereo vision.
3. Although in the category of jitter effect, **K** was ranked the biggest, and **N** the lowest, there appeared to be no significant effect in our 3D tracing task.
4. Subjects reported that **K**, although it has the biggest jitter effect, appeared to be the most responsive, while **G** was ranked second in responsiveness.

4 Psychological testing on latency effects in a real application

The goal of the experiment is to evaluate the effect of latency on human beings in a real VR application. The hypothesis of the experiment referred to as our object-following hypothesis is that once latency is involved in a system for walking through a building, subjects can follow a moving target better with latency compensation than without it.

In our observation, subjectively, we can indeed feel significant improvement in latency with grey-system compensation when wearing an HMD. However, for getting more comprehensive data on the latency effect in an HMD-based virtual environment, formal experiments and analyses on a real application cannot be missed. In these experiments, there are two major dependent variables: latency without compensation and latency with grey-system-based compensation. The *SpaceWalker* system for building walkthrough is adopted as the experimental platform (Fig. 6).

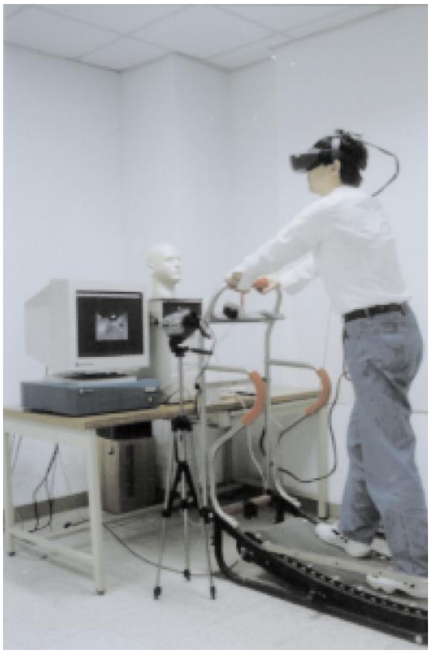


Fig. 6. The *SpaceWalker* system for a building walkthrough

The *SpaceWalker* is a polygon-based walkthrough system. The current system supports the interactive walkthrough of a building model comprising 40 180 triangles. Thanks to the visibility test to minimize the number of triangles for rendering, the update rate can reach 15–20 frames/s on an SGI Indigo^2 Extreme graphics workstation. Even though the update rate is enough to work with, it still has an overall latency of about 300 ms. The tracker contributes approximately 70 ms; the rendering pipeline, 80 ms; and the LCDs with relatively low refresh rates in the HMD contributes 150 ms. Currently, the system can be operated by three kinds of input/output devices (1) the 3D tracker to an HMD, (2) the external buttons for left and right turn, and (3) a shaft encoder-based treadmill system for measuring the walking speed.

4.1 Experimental set-up

The experiment system consists of an SGI Indigo^2 Extreme for rendering, a VR4 HMD with a Flock-of-Birds (Ascension, VT) 3D tracker for tracking head motion, and a treadmill system for measuring walking speed. During the experiment, the update rate was more than 20 frames/s, and the system latency was 300 ms. The grey system-based prediction algorithm was applied, and a prediction length of 150 ms with six historical samples was chosen. The simulated virtual space for this experiment was about 100 m^2.

4.2 Experimental design

In our experiment, the subjects were instructed to imagine participating in a guided tour in a building with wearing a HMD. The guideline trajectory, which simulates the real head movement in a tour, was prerecorded. During the experiment, there was a tetrahedron object that moved according to the prerecorded trajectory. The subjects were instructed to follow the moving object as closely as possible so that the object could be kept to the front of the HMD. Figure 7 shows a snapshot of the experiment. The system automatically records the subject's performance in terms of the off-screen rate. The off-screen rate is defined as the percentage of time that the tetrahedron object was not located on the screen for each trial, i.e., the subject could not follow the guideline, thus the object was not located on the screen.

Volunteer subjects were invited to participate in the experiment at their own walking pace and to use the *SpaceWalker* system with two virtual environments: VE1, the system without latency compensation (thus the time lag was still 300 ms), and VE2, the system with grey-based latency compensation (thus the effective latency was less than 150 ms).

4.3 Procedure

We invited six volunteer subjects to participate in our experiment. All subjects were able to navigate the virtual building, which was well controlled by a steerable treadmill with turning buttons. Each subject was given two trials of experiment: one trial was presented under VE1 and the other was generated from VE2. For three of the subjects, the experimental order was VE1 first and then VE2. For the others, the order was reversed. For each trial, the subjects were told to do his or her best to follow the moving object by using the treadmill, turning the buttons, and moving his or her head. After briefing, pressing a specific key signaled the start of each trial. A beep signaled the end of the trial. Each subject was trained to "walk" in *SpaceWalker* until additional training did not result in any significant improvement, usually within 6 min.

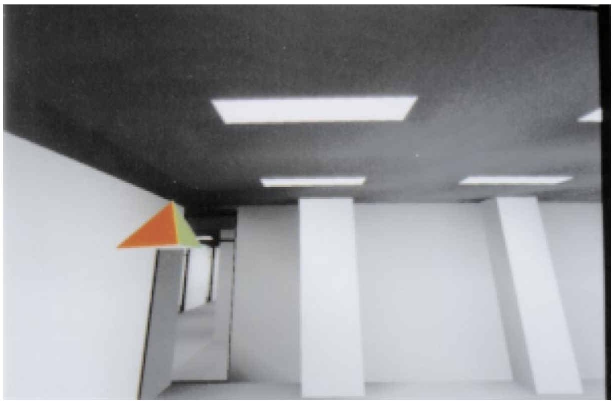


Fig. 7. A photo of our experiment. The tetrahedron object near the door is the object to be followed (a tour guide)

4.4 Results

Table 6 shows the off-screen rates in the tour guide-following task for each subject. The results corroborate our object-following hypothesis, with $t(5) = 5.13$, $P < 0.01$. Furthermore, by examining the mean rates for VE2 (with latency compensation), the average off-screen rate is 8.8%, while for VE1 (without latency compensation), the average off-screen

Table 6. Off-screen rates for the object-following experiment

Subject	VE1 (percentage)	VE2 (percentage)
1	15.0	7.8
2	19.1	9.9
3	24.5	7.1
4	19.6	8.1
5	27.3	11.6
6	14.6	8.5
Mean	20.0	8.8

$t(5) = 5.13,\ P < 0.01$

rate is 20%. Since this object-following task is similar to following a tour guide in the real world, we were able to determine that, with latency compensation, the human capability is enhanced by more than 120%.

5 Conclusions

In this paper, we have presented experiments on latency and compensation methods both quantitatively and subjectively. In our 3D target-tracing task involving eight subjects using their head motion to trace a flying target in random motion, we have found that two prediction methods, Kalman filtering (not inertia-based) and the grey system prediction, are significantly better than the method without prediction. The first two methods are equally good in a formal experiment. We also designed an experiment for a real-world application: a walkthrough in a building following a tour guide. The results again confirm our hypothesis, and an improvement of 120% was observed.

Acknowledgements. This project was partially supported by the National Science Council, project numbers NSC83-0408-E002-006 and NSC83-0425-E002-140.

Appendix: Predictions based on grey system theory

In the real world, the behaviors of most systems are uncertain [6, 7]. The effects of other systems on the system being monitored are also unclear. In grey system theory, a system model is established with a sequence of measured raw data that is generated by a system with unclear characteristics. The observed tracking data are used to generate a generating sequence on a grey generating space, and a grey differential model (GM) is applied to fit the generated sequence. By using the established GM, we can predict, analyze, and program the behavior of the original system.

In most cases, the tracker data observed by measuring the system are too random and are insufficient to establish a GM. Some manipulation of the tracking data is needed to get a more regular data sequence, and the sequence obtained is called the generated sequence.

One of commonly used operations for the generated sequence is called the accumulated generating operation (AGO). Let $x^{(0)}$ be the original tracking data sequence and $x^{(j)}$ be the generated sequence for $i > 0$. The AGO is defined as:

$$x^{(i)} = \mathrm{AGO}\, x^{(i-1)}\ ,\ i > 0 \tag{1}$$

$$x^{(i)}(k) = \sum_{m=0}^{k} x^{(i-k)}(m)\ , i > 0\ . \tag{2}$$

Since the $x^{(0)}$ are all positive, after applying the AGO, the generated sequence $x^{(j)}$ must be a monotonically increasing sequence, and its randomness disappears. Figure 8 shows an example that indicates the effects of the AGO. The prediction model GM can be established in an AGO domain.

Let $x^{(0)}$ be the original tracking data sequence with n samples, and $x^{(1)} = \mathrm{AGO}x^{(0)}$; assume they satisfy the following first-order grey differential model, GM(1,1), with a single variable:

$$x^{(0)}(k) + az^{(1)}(k) = b, k = 1, 2, \ldots \tag{3}$$

$$z^{(1)}(k) = \frac{x^{(1)}(k) + x^{(1)}(k-1)}{2}, k = 1, 2, \ldots\ , \tag{4}$$

which is obtained from the following differential equation:

$$\frac{dx^{(1)}(t)}{dt} + a \bullet x^{(1)}(t) = b\ . \tag{5}$$

Expand (3) with the n samples in $x^{(1)}$; that is, in the AGO domain. We obtain:

$$\begin{bmatrix} x^{(0)}(1) \\ x^{(0)}(2) \\ \vdots \\ x^{(0)}(n-1) \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\left(x^{(1)}(1) + x^{(1)}(0)\right) & 1 \\ -\frac{1}{2}\left(x^{(1)}(2) + x^{(1)}(1)\right) & 1 \\ \vdots & \vdots \\ -\frac{1}{2}\left(x^{(1)}(n-1) + x^{(1)}(n-2)\right) & 1 \end{bmatrix} \begin{bmatrix} a \\ \\ \\ b \end{bmatrix} . \tag{6}$$

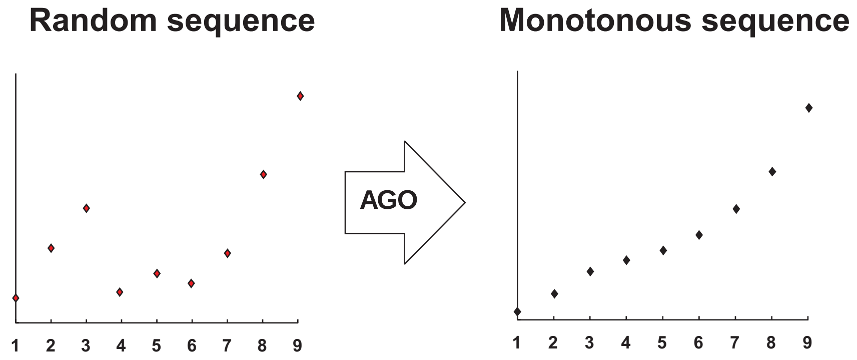


Fig. 8. The effects of the accumulated generating operation (AGO)

Let

$$Y = \begin{bmatrix} x^{(0)}(1) \\ x^{(0)}(2) \\ \vdots \\ x^{(0)}(n-1) \end{bmatrix} \quad \text{and}$$

$$B = \begin{bmatrix} -\frac{1}{2}\left(x^{(1)}(1)+x^{(1)}(0)\right) & 1 \\ -\frac{1}{2}\left(x^{(1)}(2)+x^{(1)}(1)\right) & 1 \\ \vdots & \vdots \\ -\frac{1}{2}\left(x^{(1)}(n-1)+x^{(1)}(n-2)\right) & 1 \end{bmatrix} .$$

Solving (6) with the minimal square approximation, we get a and b from the following equation:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \left[B^T B\right]^{-1} B^T Y . \tag{7}$$

By solving a, b, and the differential equation, we can get the prediction function $\hat{x}^{(1)}(k)$ for the grey system in the AGO domain:

$$\hat{x}^{(1)}(k) = \left(x^{(0)}(0) - \frac{b}{a}\right) e^{-ak} + \frac{b}{a} , \quad \text{for} \quad k \geq 0 . \tag{8}$$

Applying prediction length k in terms of the update period to the prediction function, we can get the predicted data in the AGO domain. After the inverse AGO (IAGO) defined in (9), the output data $\hat{x}^{(0)}(k)$ from the IAGO is the predicted data that we need.

Table 7. Determining the mean square error on various input data

Number of points for the prediction	Mean square error (in centimeters)
4	0.124
5	0.117
6	0.112
7	0.118
8	0.127

$$\begin{cases} \hat{x}^{(0)}(k) = \hat{x}^{(1)}(k) - \hat{x}^{(1)}(k-1) , & \text{for } k > 0 \\ \hat{x}^{(0)}(0) = \hat{x}^{(1)}(0) = \hat{x}^{(0)}(0) . \end{cases} \tag{9}$$

In the grey system theory, a variable n, which is the length of the original tracker data sequence, must be assigned as the historical input data for prediction. How many historical data as input in predicting head movement are necessary?

Through experiments, we found that using six points in a prediction with the GM is currently the best choice; we tried four points to ten points. This was done with simulations, and we picked the one that has the minimal predicted error, while considering the subjective response of someone wearing a HMD. Table 7 shows the experimental results of the mean square error on various historical input data selections. Considering the results in Table 7, prediction with six points produces the minimum mean square error, 0.112 cm, within a motion range of 25.4 cm (10 inches).

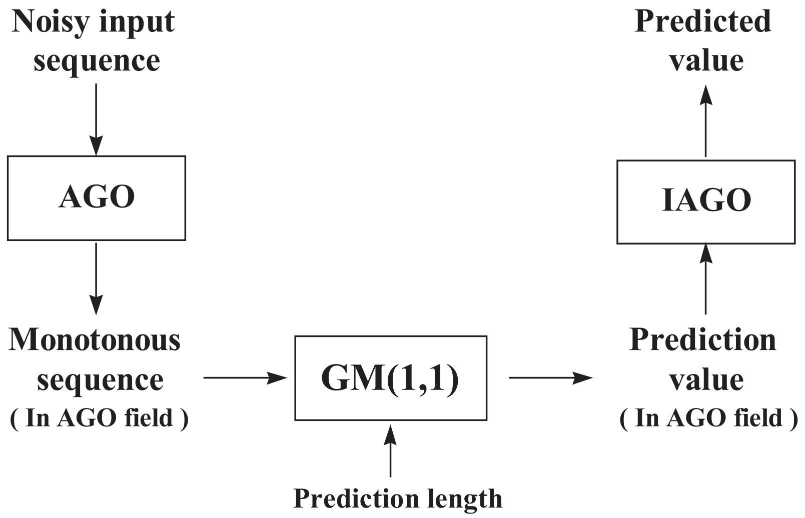


Fig. 9. Diagram of a grey system-based prediction

In our implementation, the input data of a 3D tracker consists of seven tuples (x, y, z for the position data in cartesian coordinates and Qx, Qy, Qz, Qw for quaternion orientation). The seven sequences of data are independent, and thus each has it own grey system. The following is an example to show how prediction works by using the 3D tracker data with $n = 6$ previous Qx points, to predict the seventh point. Note that Qx is one of the four parameters in quaternion algebra.

Assume the original sequence $x^{(0)}$ is:

$$\begin{aligned}x^{(0)}(k) &= \left\{x^{(0)}(0), x^{(0)}(1), x^{(0)}(2), x^{(0)}(3), x^{(0)}(4),\right.\\ &\qquad \left. x^{(0)}(5)\right\}\\ &= \{0.0355, 0.0382, 0.0398, 0.0431, 0.0478,\\ &\qquad 0.0547\}\ .\end{aligned}$$

Apply the AGO (2) to $x^{(0)}$, which is equivalent to accumulating the sequence

$$\begin{aligned}x^{(1)}(k) &= \left\{x^{(1)}(0), x^{(1)}(1), x^{(1)}(2), x^{(1)}(3), x^{(1)}(4),\right.\\ &\qquad \left. x^{(1)}(5)\right\}\\ &= \{0.0355, 0.0737, 0.1135, 0.1566, 0.2044,\\ &\qquad 0.2591\}\ .\end{aligned}$$

Note that this sequence should be increasing monotonically. Solve the differential (7) and get:

$$a = -0.093907\ , \quad b = -0.031658\ .$$

Then, the prediction function for the grey system (in the AGO domain) can be formulated as:

$$\hat{x}^{(1)}(k) = (0.0355 + 0.38953)e^{0.093907k} - 0.38953\ .$$

By (9), we predict the seventh sample point to be:

$$\hat{x}^{(0)}(6) = 0.061469$$

Figure 9 shows a diagram of a prediction based on the grey system.

References

1. Azuma R, Bishop G (1994) Improving static and dynamic registration in an optical see-through HMD. SIGGRAPH'94 Conference Proceedings, Orlando, FL, pp 197–204
2. Bloomquist K (1985) Psychological research methods: a conceptual approach. Allyn and Bacon Press, MA
3. Bozic S (1987) Kalman filter subroutine computation. Electronic Eng 24:29–31
4. Brooks FP Jr., Ouhyoung M, Batter JJ, Kilpatrick PJ (1990) Project GROPE-Haptic Displays for scientific visualization. Comput Graph 24:177–185

5. Deering M (1992) High resolution virtual reality. Comput Graph 26:915–202
6. Deng JL (1982) Control problems of grey system. Syst Control Lett 5:288–294
7. Deng JL (1989) Introduction to grey system theory. Grey Syst 1:1–24
8. Emura S, Tachi S (1994) Compensation of time lag between actual and fusion and integration. Proceedings of the IEEE International Conference On Multisensor Fusion and Integration for Intelligent System, Las Vegas, Nevada, pp 463–469
9. Foxlin E (1996) Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter. Proc VRAIS'96, Los Alamitos, CA, pp 185–194
10. Friedman M, Friedmann M, Starner T, Pentland A (1992) Device synchronization using an optimal linear filter. Proc SIGGRAPH Symposium on 3D Interactive Graphics, Boston, MA, pp 57–62
11. Jacoby RH, Adelstein B, Ellis S (1996) Improved temporal response in virtual environments through system hardware and software reorganization. Proc SPIE 2653: Stereoscopic displays and virtual reality systems III, San Jose, CA, pp 217–284
12. Kalawsky RS (1993) The science of virtual reality and virtual environments. Addison-Wesley, Wokingham, England
13. Kalman RE, Bucy RS (1961) New results in linear filtering and prediction theory. Trans ASME, J Basic Eng Series 83D:95–108
14. Lawton W, Poston T, Serra L (1994) Time-lag reduction in a medical virtual workbench. Proc BCS Conference, Leed, UK, pp 7–9
15. Liang J, Shaw C, Green M (1991) On temporal-spatial realism in the virtual reality environment. Proceedings of the 4th Annual Symposium on User Interface Software and Technology, Hilton Head, SC, pp 19–25
16. Mazuryk T, Gervautz M (1995) Two-step prediction and image deflection for exact head tracking in virtual environments. Comput Graph Forum 14:C29–C41
17. So HY, Griffin MJ (1992) Compensating lags in head-coupled displays using head position prediction and image deflection. Aircraft 29:1064–1068
18. Sutherland IE (1965) The ultimate display. IFIP 2:506
19. Wloka MM (1995) Lag in multiprocessor virtual reality. Presence 4:50–63
20. Wu JR, Ouhyoung M (1994) Reducing the latency in head-mounted displays by a novel prediction method using grey system theory. Comput Graph Forum 13:c503–c512

JIANN-RONG WU received his BS degree in Computer Science from Tamkang University, Taipei, in 1992. He received his PhD degree in Computer Science and Information Engineering from the National Taiwan University, 1996. His research interests include computer graphics, virtual reality, computer-human interfaces, and image processing. He has published over ten papers on computer graphics and virtual reality. He is also a member of the ACM.

MING OUHYOUNG received his BS and MS degrees in Electrical Engineering from the National Taiwan University, Taipei, in 1981 and 1985, respectively. He received his PhD degree in Computer Science from the University of North Carolina at Chapel Hill in 1990. Currently, he is the Director of the Center of Excellence for Research in Computer Systems, College of Engineering. He has published over 70 technical papers on computer graphics, virtual reality, and multimedia systems. He is a member of ACM and IEEE.